Author:
Kularatna, Shashitha C

Title:
Developments in Advanced Composites Skills Training and Manufacturing through Virtual Reality and an Artificially Intelligent Layup Agent

Bristol Composites Institute

Department of Aerospace Engineering

University of Bristol

Developments in Advanced Composites Skills Training and Manufacturing through Virtual Reality and an Artificially Intelligent Layup Agent

---

By

Shashitha Chamaka Kularatna

A thesis submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering, Department of Aerospace Engineering

May 2020

Word Count: 63342

Abstract

Despite being the dominant manufacturing route, hand layup of composite materials is still poorly understood. Until recently, only very little documented knowledge has existed on how the manual portion of the hand layup process was actually carried out. On a simple level, hand layup is a process whereby flat sheets of mainly pre-impregnated woven clothes (prepreg) are deformed into complex shapes using hands and support tools. It is labour intensive and hence can be prone to large amounts of variation. A large proportion of these variations can be attributed to the operator skill, layup instructions, geometry, and material. A potential workforce shortage, with the necessary skills to undertake the layup tasks, has been identified as one of the major concerns in the near term. Aligned to this is an existing and immediate need for re-skilling and/or up-skilling of the current workforce, as well as standardization of training delivery. Work presented in this thesis has attempted to address these issues by exploiting Virtual Reality (VR) technology, the concept of Gamification, and artificial intelligence techniques.

A feasibility study on the use of VR technology and gamification on composites skills/knowledge transfer was carried out on the layup of a flat carbon fibre composite panel using unidirectional prepreg material in a typical composite clean room environment. The VR system used included a smartphone based head mounted display, hand held game pad as the input device and a headphone, which outputs audio instructions to the user. The conclusion from the evaluation of the training aid was that the designed training aid, in its current form is an effective knowledge capture/transfer tool, but so far has limited use in physical skill transfer due to the use of a hand held input device. As a knowledge capture/transfer tool, it showed potential in the reduction of the learning curve of novice laminators. It was also identified that this concept can easily be adapted to layup of more complex parts with the use of appropriate alternative hardware that would enable the transfer of physical skills via hand tracking.

A second VR training simulation was designed for a 3D mould shape. The VR system used for this simulation consisted of an Oculus Rift Development Kit 2 and a Leap Motion Controller as a hand tracker. Nineteen distinct actions were identified as required to complete the layup of the chosen mould through video analysis of expert laminators

laying up the chosen mould. Relevant experiments were carried out to obtain a preliminary understanding on the suitability of VR technology as a platform for hand lamination training. Overall feedback received from the test candidates suggested that the VR training aid in its current form might not be suitable as a standalone platform for hand lamination training. However, experimental results did not fully agree with the opinions of the test candidates, since test candidates who received VR training only, performed better than the test candidates who received one to one training during their layup trials. Lack of haptic feedback was identified as a major bottleneck in using such VR systems to train laminators, especially when complex mould shapes that require in plane shear to be applied to prepreg material to complete the layup are in consideration.

Following on from the two VR related case studies, inability to predict the sequence of actions required to layup a given mould shape was identified as a major issue in designing and potentially automating the generation of such VR training simulations. Previous attempts made at using outputs from existing drape simulators to provide layup instructions were reviewed. The conclusion here was that such previous attempts have failed to provide sufficient information required for shop-floor operatives to carry out layup operations. An artificially intelligent (AI) layup agent capable of determining the ideal layup sequence for a given mould shape based on deep reinforcement learning was proposed as a solution to this issue. Three distinct case studies were carried out to explore working mechanics and limitations of the proposed AI agent. Results from the case studies suggested that, the AI agent in its current form is capable of determining the ideal layup sequence for a given mould shape, if the complexity of the mould shape is not beyond the current capabilities of the AI agent. Factors that limit the current AI agent from being able to handle mould geometries that are more complex have been identified and possible solutions to these limitations have been proposed as future work.

The work presented in this thesis has laid down the foundation and introduced new frameworks and use cases for emerging technologies in solving key issues around the hand layup process of prepreg. Both case studies carried out related to VR training for prepreg layup showed an increase in accuracy of layup >20% when compared with traditional training methods. This is a clear illustration that VR training is bound to play

a key role in training hand laminators in the near future. In addition, the AI layup agent introduced in this thesis has opened up a completely new research avenue for the development of future drape simulators that will be capable of providing optimized and numerically validated instructions for hand laminators and the design of such VR training simulations.

Dedication and Acknowledgements

The author would like to acknowledge the support of the EPSRC through the Advanced Composites Centre for Innovation and Science's Centre for Doctoral Training (Grant: EP/G036772/1) and the EPSRC Centre for Innovative Manufacturing in Composites (CIMComp) (Grant: EP/IO33513/1).

Special acknowledgement goes to Dr Carwyn Ward for helping and guiding me through every single aspect of my PhD. It must be mentioned, he has provided me with much needed moral support and motivation from the beginning to the end of this PhD and without his support, this PhD would never have been completed.

Thank you to Professor Kevin Potter for his input in moulding my PhD.

Thank you to Professor Paul Weaver who is the director of the Centre for Doctoral training for always encouraging and demanding high quality research from all students under him, which has been a major inspiration for me.

Thank you to Mike Elkington, Dominic Bloom, Helene Jones and Dennis Crowley for their input to my PhD project and guiding me in the right path when needed.

Thank you to all the technical staff at ACCIS including Julie Etches, Yusuf Mahadik, Guy Tricks and Ian Chorley for all their support during my PhD.

Thank you to all the CDT 14 and 15 cohort members who helped me carry out many of my feasibility studies.

Special thanks to Sarah Hallworth, who has helped me in many occasions especially related to administrative work of my PhD.

Thank you to Manu Mulakkal and Jamie Hartley for being great friends and motivating me in times of need during my PhD and helping me carry out certain experiments when needed.

Finally, very special thank you to both my father and mother, who despite being far away from me during my PhD have done unspeakable things for me to make sure I get through my PhD. I want to dedicate this PhD to both of you.

Author's Declaration

The accompanying dissertation entitled 'Developments in Advanced Composites Skills Training and Manufacturing through Gamification and an Artificially Intelligent Layup Agent' is submitted in support for an application for the degree of Doctor of Philosophy in Engineering to the University of Bristol.

The dissertation is based upon independent work by the candidate. All contributions from others have been acknowledged in the dissertation. The supervisors' contributions were those normally made in a British University. The views expressed within the dissertation are those of the author and not of the University of Bristol.

None of the work described has been, or is being, submitted for any other degree or diploma to this or any other institution.

Signature: ……

Name: Shashitha Chamaka Kularatna

Date: ……

Table of Contents

List of Figures

List of Tables

Notation

A/a = Action

AFP = Automated Fibre Placement

AI = Artificial Intelligence / Artificially Intelligent

$\mathbf{A}_t^*$ = Action with the Highest Estimated Value at Time Step 't'

ATL = Automated Tape Laying

b = Bias

C = Weight Swap Steps

CNN = Convolutional Neural Network

CPU = Central Processing Unit

E = Expected Value

FEM = Finite Element Modelling

FRP = Fibre Reinforced Polymers

G = Expected Return

GPU = Graphic Processing Unit

HMD = Head Mounted Display

L (.) = Loss Function

LPP = Left Pinky Position/ Left Palm Position

LWP = Left Wrist Position

MDP = Markov Decision Process

P/p = Probability

PJN = Pin Jointed Net

q_π = Action-Value Function for Policy π.

Q_t (a) = Estimated Value of Action 'a' at Time Step 't'

R/r = Reward

ReLU = Rectified Linear Unit

RPP = Right Pinky Position/ Right Palm Position

RTM = Resin Transfer Moulding

RWP = Right Wrist Position

S/s = State

T/t = Time Step

v = Momentum

v_π = State-Value Function for Policy π

VARTM = Vacuum Assisted Resin Transfer Moulding

VFP = Virtual Fabric Placement (Drape Simulation Program)

VR = Virtual Reality

w/θ = Synaptic Weight

α = Learning Rate

ϒ = Discount Rate

θ/w = Synaptic Weight

θ_L = Locking Angle

π = Policy

φ (.) = Activation Function

CHAPTER 1: Introduction to Composites Manufacturing and Understanding Research Motivations

1.1 Composite Materials

Any material that is made from more than two distinct constituents is defined as a composite material. By this definition, many materials are effectively composites. This is particularly true of many biologically natural materials, which are more often than not made out of more than two or more different constituents. In many cases, one of these constituents is strong and stiff and is in an elongated form. This particular constituent is called the reinforcement. The reinforcement can also be found in particulate form. The reinforcement is embedded in another softer constituent, which forms the matrix. A common example is wood, which is made out of fibrous chains of cellulose molecules embedded in a matrix of lignin. While the reinforcement provides strength and stiffness to a composite structure, the matrix holds the reinforcement in place while transferring any load applied to the structure between the fibres and supporting the structure under compression. Due to the structure of composite materials, more often than not, they exhibit anisotropic properties – that is to say, their properties when measured in different directions vary significantly [1].

It is clear from the definition of a composite material that there exist many different types of composite materials. However, in this thesis, only a particular group of composite materials named ‘Fibre Reinforced Polymers (FRP)’ will be considered. Therefore, the use of the word ‘composites’ will exclusively refer to fibre reinforced polymers from this point onwards, unless otherwise stated.

1.2 Fibre Reinforced Polymers (FRP)

The essence of FRP technology is the ability to place the appropriate type of strong and stiff fibres within the appropriate type of matrix based on the application in the correct place, orientation and volume fraction to achieve desired structural properties [1].

Glass fibre reinforced polymers (GFRP) are the most commonly used form of composites. It is also the cheapest form of composite materials and is used for bathtubs, car body panels, boat hulls, motorcycle helmets, and various other day-to-day applications [1]. But, when it comes to weight critical high-performance applications such as Formula 1 racing cars, civil and military aircraft components and sporting equipment, the reinforcement that is preferred over glass is carbon fibre which has higher strength and stiffness compared to glass. However, these superior properties come at a higher monetary cost.

Carbon fibre and composites in general have been replacing engineering metals as the material of choice in a variety of applications. The higher monetary cost in switching from metals to composites in high performance applications is in general offset by gains in performance and long-term operational cost savings. More than 50% of Airbus's A350 XWB and Boeing's 787 Dreamliner structures are made of composite materials leading to dramatic decreases in the weight of these aircrafts and their fuel consumption [2], [3]. McLaren revolutionised the production of racing cars in 1981 with the production of Formula 1's first carbon fibre chassis, today most of the chassis of Formula 1 racing cars, suspension, wings and engine cover are built with carbon fibre [4]. While high performance racing car manufacturers were quick to embrace the use of composite materials, this same trend did not extend in to the large-scale automotive industry until recently. The automotive industry remained largely based around metallic structures primarily due to economic reasons. Around 12% of greenhouse gas emissions in the EU is because of the automotive industry [5]. Legislations such as the Kyoto protocol has been put in place because of increasing concerns about these emission rates [6]. This has led the automotive industry to search for more sustainable solutions based around composites. In the electric car sector, where weight and efficiency are even bigger

drivers due to heavy weight batteries with low energy capacities, use of composites has been more readily accepted. The BMW i3 and i8 are examples from the electric car sector [7].

The main difference for varying levels of adaptation of composites in the car industry is driven by different production volumes. While Formula One companies produce a handful number of cars per year, about 70 BMW i3 cars are produced per day and when it comes to more commonly used cars such as the Ford Focus, around 3000 cars are manufactured per day [8], [9]. The main challenge that is preventing the automotive sector from widespread adaptation of composite materials is the inability of current composite manufacturing methods to meet high production rates while maintaining quality standards and making sure costs are kept low to remain competitive in such a price driven industry [10]. While the work presented in this thesis tries to provide possible solutions to some of these issues, before presenting said solutions, it is certainly worthwhile reviewing the state of the art of composite manufacturing processes.

1.3 Manufacture of Composites

Unlike engineering metals that are usually obtained in block, rod, plank, or sheet form for use as raw material in machining, moulding, and joining operations to manufacture the final component, when it comes to composites, in general, both raw material and the final component are made in the same manufacturing process [11]. Composite manufacture is made inherently complex by the fibrous nature and low strain capability of reinforcement material used [12]. While there exist many composite manufacturing methods, starting from the early application of glass fibre to the latest carbon fibre Airbus A350 XWB, for the purpose of this thesis, only those manufacturing processes used for high performance continuous fibre composites are reviewed in this section. In depth descriptions of manufacturing processes such as spray layup, injection moulding, and Sheet Moulding Compound (SMC) that are used to produce short fibre composites can be found in various composite manufacturing text books such as [13] & [14].

1.3.1 Wet Layup

Wet layup is a process where dry reinforcement is placed on to a mould and liquid epoxy resin is applied to the reinforcement, usually by hands. Wet layup is the simplest and most versatile composite manufacturing method to date. Despite being one of the oldest and first composite manufacturing methods, wet layup is still widely used due to its many advantages such as being extremely versatile, requires relatively low capital investment (can be achieved with as little as a well ventilated room, a mould and a brush) and prior knowledge, and being highly economical for prototype production and low volume short production cycles [11]. However, the volume fraction of the fibres and therefore structural properties of parts produced with wet layup tends to be low and variable. Parts made via wet layup typically are not weight critical or cost driven. Bathtubs and boat hulls are prime examples [1].

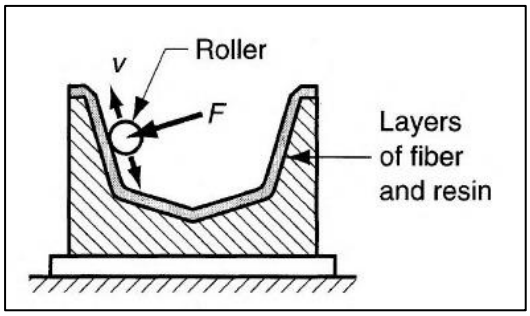


Figure 1.1 – Wet layup process (v – Roll Direction, F – Roller Force Direction)

1.3.2 Prepeg Layup

Layup of prepreg is a refined form of wet hand layup, which provides higher performance parts with superior and more consistent fibre volume fraction. Due to this superiority in structural properties, virtually all high-performance composites are manufactured using prepreg [11].

“Prepreg” is the common term used for reinforcement that has been pre-impregnated with a resin system. Prepreg comes in two forms, ‘unidirectional’ and ‘woven’. While fibres run in a single direction in unidirectional prepreg, they run in two directions weaving in and out of each other that are perpendicular to each other in woven prepreg.

The resin in prepreg holds the fibres in place and makes it significantly easier to handle the material when compared to handling raw fibres and resin separately. Both forms of prepreg are draped into shape on a mould by hands. This means that prepreg layup allows for the creation of parts with a wide range of complex shapes. However, not all prepreg layup is manual. Automation of prepreg layup exists for parts with simple shapes. This will be covered later in this section.

In prepreg layup, each ply must be laid down separately. This makes the process labour intensive. However, it also allows each ply to be inspected once it is placed where required, which allows the creation of high-quality parts [12]. This process can also be easily adapted to new parts and design changes and has relatively low start-up costs. Some of its drawbacks are low production rates and high material and labour costs. However, for short production runs, prepreg layup can become very cost effective due to its adaptability to design changes [15]. In addition, just like any other manual process, parts produced by prepreg layup are susceptible to discrepancies due to inherent human variation. Even at the presence of these drawbacks, the ease of adaptability and superior quality of parts produced through this process means that it is crucial to the composite industry and remains the main manufacturing route for many applications in the composite industry [16]. Prepreg layup will be revisited in more detail later in the Introduction, and hence is not further reviewed here.

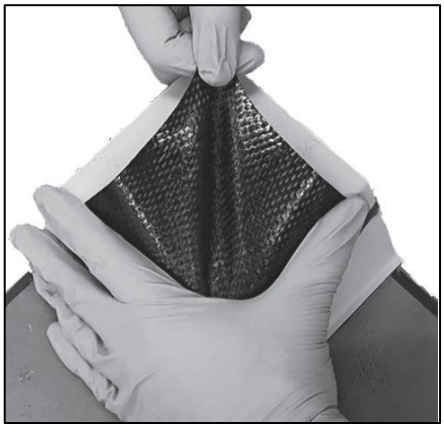


Figure 1.2 – Hand layup of woven prepreg: An example [16]

1.3.3 Compression Moulding

In compression moulding, composite plies are stacked together, and matching male and female mould halves are used to form the stacked set of plies in to the desired shape. The stacked set of composite plies, which is called the charge is placed on top of the female mould and the two half moulds are then closed under high pressure forcing the charge to flow to fill the mould. The mould is then heated to ensure that cross-linking is completed. Once the part is dimensionally stable, it is demoulded [11].

Compression moulding is most cost effective for long production series, and hence has had the greatest success in the automotive industry. One of the most common applications is the manufacture of vehicle body panels [11]. While compression moulding is suitable for high volume production, large investments are required to set up the infrastructure and tooling. Therefore, long production runs may be required to gather the initial investments [17]. In addition to this, compression moulding is usually limited to thermoplastic resins, which melt and turn in to fluid under heat allowing the composite plies to easily deform and slide against each other and single or very lightly doubly curved component shapes [17].

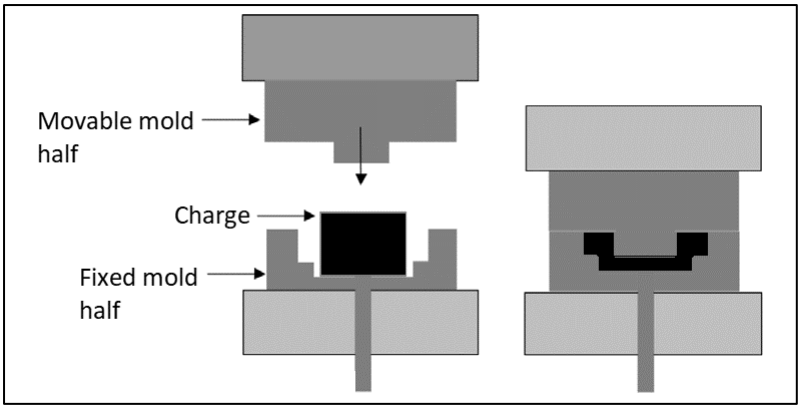


Figure 1.3 – Compression moulding process

1.3.4 Resin Transfer Moulding (RTM)

RTM is the most common liquid moulding technique for manufacturing structurally capable composite structures. Its popularity is due to its ability to produce large, complex, and highly integrated components while keeping the capital cost relatively low

[11]. RTM was first used for composite manufacturing in the 1980's, and was driven by the automotive industry that was looking for high volume production of net shaped structural parts. While injection and compression moulding of short fibres were capable of producing net shaped structural parts at high volumes, the required structural properties were not achieved with the use of short fibres. This led to the use of woven or stitched fibre preforms, which were placed inside a closed net shaped mould, which was injected with low viscosity resin under high pressure, which filled the empty spaces between the fibres [12].

The placement of preform in the mould is a labour intensive task, which requires less skill than layup of prepreg [11], [18], [19], [20]. The closed mould allows the achievement of fine thickness and shape control, but this can also make it difficult to detect defects [21]. Few major drawbacks of RTM have prevented it from dominating composite manufacture within the industry. RTM has been limited to long and high-volume production processes due to high costs involved in constructing the moulds and other equipment required for the process [22]. The RTM process is very labour intensive and requires a highly skilled workforce, especially to load the mould with the preform. The quality of the preform also greatly effects the quality and structural properties of the final component produced [14].

There are also issues involved with the injection process, such as its unpredictability and difficulty of control. One of the most common problems is the displacement of fibres around injection points [21], [23]. Usually before full-scale production can begin with RTM, numerous prototypes are required to get mould design features such as pressures and resin injection ports right. If resin injection is not done right, the cured part will contain voids and dry spots [22].

Vacuum Assisted Resin Transfer Moulding (VARTM) is a variation of RTM, where half of the RTM mould is replaced with bagging material. Resin is supplied to one side of the mould, while a vacuum applied to the other side of the mould draws the resin through the mould wetting the preform. Flow of resin across the part is enhanced by placing a layer with high permeability on top of the reinforcement [24], [25]. The resin filling

process can be highly variable, resulting the strength of laminates produced via VARTM to be highly dependent on the process quality. Structural properties of parts produced via VARTM in general tends to be inferior to parts produced via prepreg layup [26].

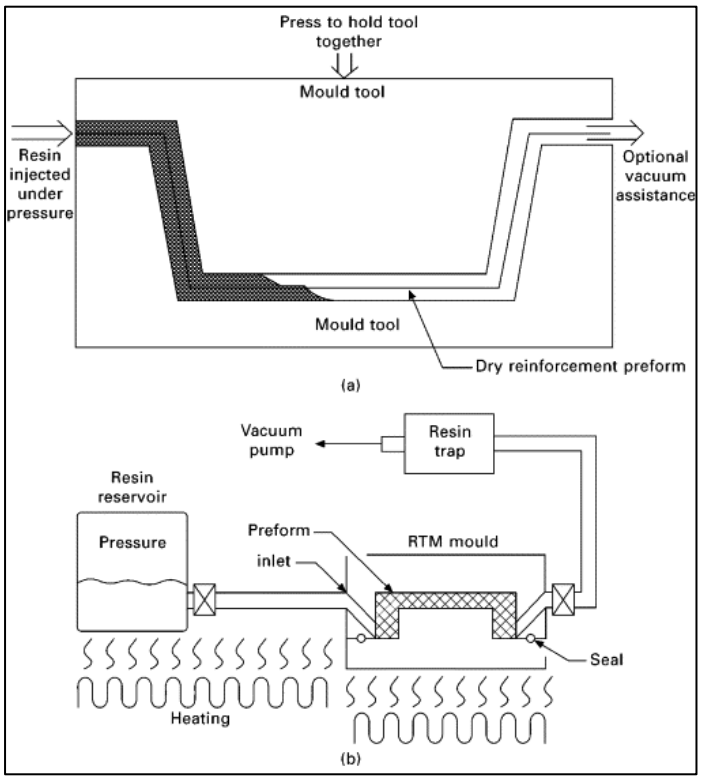


Figure 1.4 – (a): Resin transfer moulding process, (b) Vacuum assisted resin transfer moulding process [27]

1.3.5 Pultrusion

Pultrusion is fully automated and is used to produce continuous fibre reinforced profiles with constant cross section [12]. Pultrusion is the most cost-effective method for high volume production of composites. Pultruded composites are used in every conceivable application spanning from electrical and civil engineering to sports and medicine [11]. Its major drawback is embodied within its definition, which is only parts with constant cross-section can be manufactured using this method. As production of such parts with constant cross section is of no interest to the work presented in this thesis, Pultrusion will not be discussed any further. More information on Pultrusion can be found in textbooks such as [11] & [12].

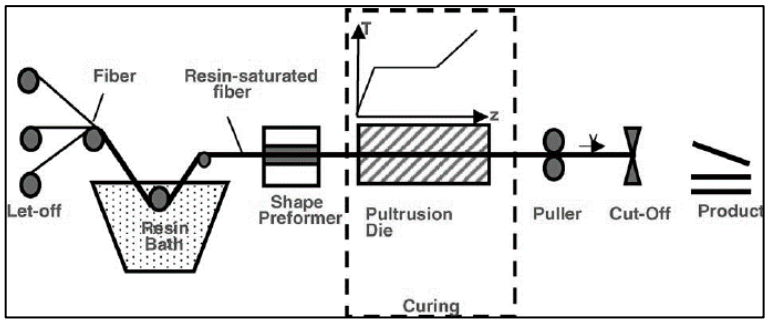


Figure 1.5 – Pultrusion process [28]

1.3.6 Filament Winding

Filament winding is a highly efficient and automated process that allows precise placement of yarns that are impregnated with resin on to a rotating mould. Similar to pultrusion, filament winding is highly cost effective, but is limited in the shapes of structures it can produce. Geometries that can be produced are limited to convex and closed geometries that can be rotated. Pressure vessels, pipes, launch tubes, and other rotationally symmetric components are common examples of structures that can be manufactured via filament winding [11].

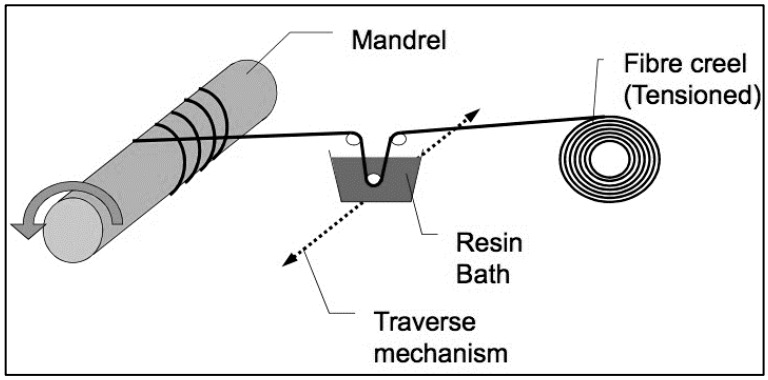


Figure 1.6 – Filament winding process [29]

1.3.7 Automated Tape Laying (ATL) and Automated Fibre Placement (AFP)

Both ATL and AFP are attempts made at automating the process of prepreg layup. In both these methods, rather than using large sheets of prepreg, unidirectional strips of prepreg are laid down by a robotic arm. While prepreg tapes are the current standard

material used in ATL/AFP, there are developments in using dry fibre tapes [30]. In [10], an in-depth review of both ATL and AFP can be found.

ATL is limited to single curvature and flat components due to the use of wide tapes (~100mm). In AFP, thinner tows of width around 6mm are used. By steering tows in plane during layup, AFP enables in full, the concept of variable-angle tow composites [31]. In contrast to non-steered laminates, this vastly improves properties such as stiffness and buckling [32]. However, In-plane bending of the tow element is required to achieve a curved tow path. Therefore, in principle, deformations such as local buckling and thickness changes are unavoidable. Therefore, a minimum curvature of the tow path must be kept to reduce these local defects [33].

When it comes to out of plane deformations, the working principle of AFP is that the double curvature of the part to be laid up is at a much larger radius compared to the individual tows with the previously quoted width of 6mm. The result of this phenomenon is that the area covered by every single tape is essentially singly curved. This also means that no significant shear deformation is required.

In comparison to Hand Layup, AFP can provide similar or better accuracy in tow placement. Being a fully automated process means that it can also provide faster layup rates. However, AFP can only handle a limited set of geometries. “Coriolis” an AFP manufacturer has quoted 20^{o} to be the steepest angled ramp their AFP machines can successfully layup [22]. Due to the existence of a minimum radius of tows used, severe double curvatures can be an issue for AFP machines. Huge initial investments that are required on AFP machinery is another major drawback.

Figure 1.7 – Schematic of an automated fibre placement machine in use [34]

1.4 Primary Steps in Composites Manufacturing Processes

Despite the presence of multiple composite manufacturing processes as explained in Section 1.3, there are broadly 7 steps in any composite manufacturing process [35]:

Step 1: Storage

The materials used are removed from storage. In the case of catalysed resin, either infused into the fibres or stored separately, it must be kept at low temperatures.

Step 2: Cutting

The fibres are typically stored in rolls as fabric and need to be cut to size/shape. In the case of some more automated methods this occurs after layup.

Step 3: Kitting/Ordering/Stacking

The fibres act as reinforcement, so their orientation is crucial to the strength of the component. For manually laid-up components the plies of fabric are stacked into a “kit” to ensure they are in the correct sequence.

Step 4: Tool preparation

The specifics of tool preparation depend on the tool material, the component material and the manufacturing route. What is common to all processes is that some form of

barrier/release agent is applied to the tool to prevent the polymer resin from bonding to the tool, meaning the tool and component cannot be separated without damage.

Step 5: Layup

Layup is the process of applying the fibres to the tooling substrate. This stage, for manual layup, has the least documented process control [36].

Step 6: Cure

Once the part is laid-up it must be cured to lock the fibres in place and harden the polymer matrix. The cure specifics are determined by the chemistry of the resin and the chosen method of curing/manufacture.

Step 7: Finish

Following cure there are some steps for finishing the part. These include: removing vacuum bags, trimming edges, painting/coating/sealing processes.

1.5 The Ideal Composite Manufacturing Process

The presence of such a variety of composite manufacturing methods as evident from the above section already suggests that the choice of the ideal composite manufacturing process for a particular part must be based on a variety of factors. All manufacturing processes discussed above have their distinct advantages and disadvantages. Choosing the correct manufacturing process is critical in making sure that the composite manufacturing process is successful. Size of the part, geometrical shape of the part, production volume, and available capital and infrastructure are some of the factors to be taken in to account when selecting the ideal manufacturing route. Figure 1.8 provides a comparison between few of the composites manufacturing processes in relation to choosing the ideal process.

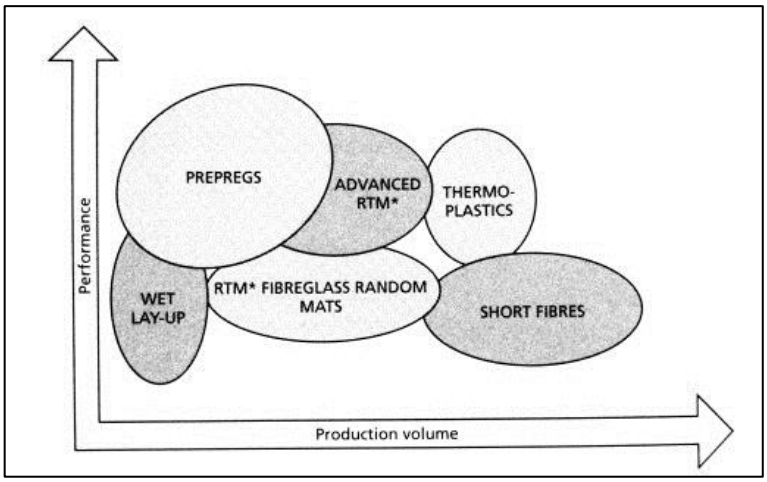


Figure 1.8 – Comparison of composite manufacturing processes [37]

Consider an aircraft wing spar with a span of around 5m. Being a component of an aircraft means, it requires being of the highest quality and structural properties. This immediately rules out the use of wet layup and VARTM. Due to its size (5m), use of RTM or compression moulding becomes not so desirable due to the cost of moulds and large forces required to maintain pressure over such a large area. The two logical choices are AFP or Hand Layup. Wing spars usually have simple geometries and production life cycles are bound to last over many years. In addition, production rates are moderate, where 1 to 10 wing spars to be produced per day. Therefore, it is reasonable to invest on an AFP machine [38].

In contrast, inspection panels of aircrafts that consist of a sandwich core, which are secondary components pose a different manufacturing challenge to the previously discussed wing spar. Inspection panels are still load carrying structures that require high quality structural properties. Due to this reason, wet layup and VARTM are ruled out once again. Production volume requirements may hover around 2 to 3 panels per aircraft per day. Subtle variations in design between inspection panels mean that the use of separate RTM or press moulds to match these designs may make these processes expensive and complicated. Complex shapes created due to the presence of the sandwich core may mean that this is beyond the capabilities of AFP, leaving hand Layup the ‘ideal’ process [38].

Similar to the case discussed above, there will be various other cases where Hand Layup of Prepreg is the 'ideal' manufacturing route. Hand Layup of prepreg is unique in its ability to manufacture parts with complex geometries to high quality with very little initial investments. It is this very uniqueness that makes hand layup an important component of the composite manufacturing industry. The head of operations for Airbus – Miguel Morell, has mentioned in a presentation at ICCM19 in 2013 that although the company needs to bring in more automation in the future, hand layup is still the proper way to manufacture selected complex components [39]. The suitability of hand layup increases furthermore, when it comes to shorter or one-off production cycles such as Formula 1 car body panels. There is also a push towards reducing the part count in weight critical components to avoid weight penalties associated with mounts or brackets [40]. Such requirements lead to increasingly complex geometries, which makes the use of AFP systems even more difficult and possibly making processes similar to hand layup more favourable.

1.6 Difficulty in Hand Layup of Complex Parts

Before answering this question, it is essential to understand few basic principles. Surfaces with single curvature such as cones or cylinders are categorized as 'developable'. I.e. they can be formed using a flat sheet with no in plane deformations. However, forming of any doubly curved surface require a certain amount of in plane deformation [41]. Materials that can exhibit large plastic deformations and have isotropic mechanical properties can be formed on to doubly curved surfaces with relative ease by using either a vacuum or a stamp former [42]. Thin plastic and metal sheets are examples of such materials. However, forming of composites over a doubly curved surface presents a more complex problem. This is because materials such as carbon and glass fibre inherently possess high stiffness and low strain to failure. This prevents them from forming over doubly curved surfaces in a similar manner to thin plastic or metal sheets [43]. However, when fibres are arranged in the form of a woven cloth, the material as a whole gains flexibility in terms of in-plane scissor type shear. Woven cloths can be deformed and 'stretched' or 'contracted' in opposite directions.

Even though the major form of deformation displayed by woven cloths is this scissor type shear, deformations such as tow slippage can also be observed to a certain extent.

1.7 Automation of the Layup Process

The most significant direct cost associated with composite layup is the labour cost, as it is still primarily conducted by highly experienced and qualified trades-people [44]. Development of automated systems will be aided by the breakdown, understanding and knowledge capture of layup processes. Lack of process understanding and "The human element" has been identified in [45] as main sources of errors within a manufacturing process. Current automated processes are unable to layup many of the complex shapes that are being laminated manually. For example, if an aircraft is considered, AFP has been used to successfully lay down geometrically simple but large components such as fuselages. However, AFP cannot be used to layup complex parts such as seats, luggage holders and other interior components with complex shapes. Due to this reason, a significant percentage of certain AFP parts are still laid up manually by humans. A review on most recent attempt at automating the layup process or parts of it can be found in [46].

1.8 Improving the Hand Layup Process

Despite being the dominant manufacturing route, hand layup of composite materials is still poorly understood. Until recently, only very little documented knowledge has existed on how the manual portion of the hand layup process was carried out. According to [16], a review of the leading textbooks on composites gives very little information on hand layup, with their authors describing the layup process using a single sentence. Despite its importance, the basic working mechanics of the hand layup process has changed very little in the past 30 years [16]. In addition, most recent research has given priority to refining automated techniques such as AFP [10] over hand layup.

On a simple level, hand layup is a process whereby flat sheets of mainly pre-impregnated woven clothes are deformed into complex shapes using hands and support tools. It is labour intensive and hence can be prone to large amounts of variation. A large

proportion of these variations can be attributed to the operator skill, layup instructions, geometry, and material.

In most instances of components made by hand layup, there will be an accompanying Manufacturing Instruction Sheet (MIS), which details the manufacturing process. Whilst an MIS will dictate the sequence of plies, the specifics of how to drape the fabric ply over the tool surface to achieve this is left to the operators' discretion. Table 1.1 shows an example of this, with a column for a signature or stamp to certify conformance [35].

The issue with leaving the layup process up to the users' discretion is that there are multiple ways to drape a ply over a surface. Different starting points and paths of sticking or smoothing the material can lead to different patterns of reinforcement and thus effectively different parts. There are also multiple ways to achieve the same reinforcement pattern, so there is isn't one "right answer". These introduce variability and whilst this variability is typically small, it has design implications as strength "knock-down" modifiers are applied to account for variability in manufacture. If it could be shown that parts could be manufactured reliably, it may be possible to lessen these strength reduction factors to make more efficient use of material [35].

Table 1.1 - Example of a manufacturing instruction sheet for hand layup of prepreg

Ply Number	**Orientation**	**Completed**
1	45°	
2	0°	
3	90°	
4	45°	

The work presented in thesis is a detailed description of novel attempts made at addressing the above-mentioned issues around the hand layup process. Before presenting said attempts, it is worthwhile to review the most recent attempts made at understanding and improving the hand layup process.

The first known detailed study of the techniques used by laminators during the hand layup process was published in 2015 by M. Elkington [16]. In this study, 4 participants were made to layup 15 different mould shapes that represented features found on common composite parts. These attempts were all recorded and video analysis techniques from the field of ergonomics were used to identify eight distinct techniques used by the laminators. A systematic approach to layup was identified through this study that revealed strong links between specific mould features and techniques used by the laminators. The author also claims that this study could enable the first step toward building a design for manufacture knowledge base around hand layup. Further claims are made on the use of this study for improvement of layup training and future automated layup solutions.

H.V. Jones carried out the first known investigation of a standardized tool for composite layup in 2015 [47]. In this study, personal handheld tools used by laminators to form advanced composites were investigated. The author has highlighted the lack of formal knowledge surrounding the existence and use of these tools and has explored paths toward a possible standardization of layup tools. Layup trials have been carried out to test the use of a prototype tool named the “Dibber”. This study has also shown that the development of a standardized tool could benefit in improving skills and training within composite layup.

In [48], D. Bloom has carried out a study of the effect of prepreg properties on the hand layup process. The properties of prepreg considered in this study were shear behaviour, tack and flexural rigidity. Behaviour of prepreg is highly dependent on time/rate and temperature due to its viscoelastic nature. Therefore, these properties have been measured in a temperature and humidity-controlled composites clean room under adjusted test parameters to represent the forces and rates the material might experience via the hands of a laminator. A collection of commercial prepregs was laid up by experienced laminators on increasingly difficult mould shapes, and time taken to completion of layup was recorded. The author has reported a positive correlation between shear angle and layup time. A significant difference (up to a factor of 2) in time

to layup between different prepreg materials was also recorded. However, the author has failed to investigate an order effect in this study. To be more precise, layup time could be affected by the same laminator laying up the same mould multiple times with different prepreg material. This could lead to a reduction in layup time irrespective of material properties. On the other hand, since the author has quoted that each laminator had to complete 40 layup trials, layup time could increase towards the latter layups due to laminator exhaustion.

In [49], M. Such has identified the need to improve real-time guidance of laminators in trying to improve and standardise the lamination process. This issue has been addressed by proposing a novel and low-cost system enabling laminators to be guided in real-time based on a predetermined set of instructions. This system consisted of an overhead projector that projected layup instructions directly on to the mould. Microsoft Kinect^{TM} was used to track the laminators hand movements in real-time and adjust the projected instructions accordingly. This system is quoted to reduce the variability in the expected global shear of a laid up ply by addressing the knowledgebase of a skilled laminator, improve standardisation of components laid up by different laminators and reduce the learning curve of new laminators. This particular project is currently being further developed and investigated under the name of LayupRITE by D. Crowley and C.Ward [50].

When laying up composite parts with complex geometries using woven prepreg, more often than not, the required shear pattern of the prepreg tends to be non-intuitive. To tackle this issue, computer simulation programs such as Virtual Fabric Placement (VFP) [51], and Unity Drape Simulator [52] have been developed. In [53], a review of how woven reinforcement are conventionally used in hand layup can be found. It discusses about the use of kinematic modelling to inform laminators during the process of hand layup. The authors have further extended their work in to providing a strategy that allows the output information from kinematic modelling software to be used to produce manufacturing instructions that can be provided to laminators during the layup process.

This work was completed in the year 2006 and is recognised as the first step in developing such tools. [54] & [55] are two more examples of similar studies.

1.9 Summary and Understanding Research Motivations

In this chapter, the concept of composite materials and its subgroup "Fibre Reinforced Polymers (FRP)" were introduced. Most common manufacturing methods for high performance FRP and their distinct advantages/disadvantages were discussed. Hand layup of prepreg was identified as the dominant form of FRP manufacture. A review of work done in improving the process of hand layup of prepreg was discussed and key areas to focus on further improvements were identified. In the next few chapters of this PhD thesis, these identified areas will be further discussed and novel solutions to some of these issues will be explored.

Figure 1.9 attempts to justify the motivations behind the research presented in this thesis. According to this figure, a fully automated solution (S4) for the layup of complex composite parts will in theory solve all current issues around the manual portion of the layup process. However, it was established in this chapter that we do not currently have such a solution. Therefore, until such an automated solution is found, solutions to existing issues around the manual portion of the layup process must be pursued. Apart from the fully automated solution (S4), there are three other solutions suggested in Figure 1.9 (S1 – S3). S3 is already being pursued through the LayupRITE project as explained in section 1.8 of this chapter. Therefore, the objective of the work presented in this thesis is to pursue solutions 1 and 2 (S1 and S2).

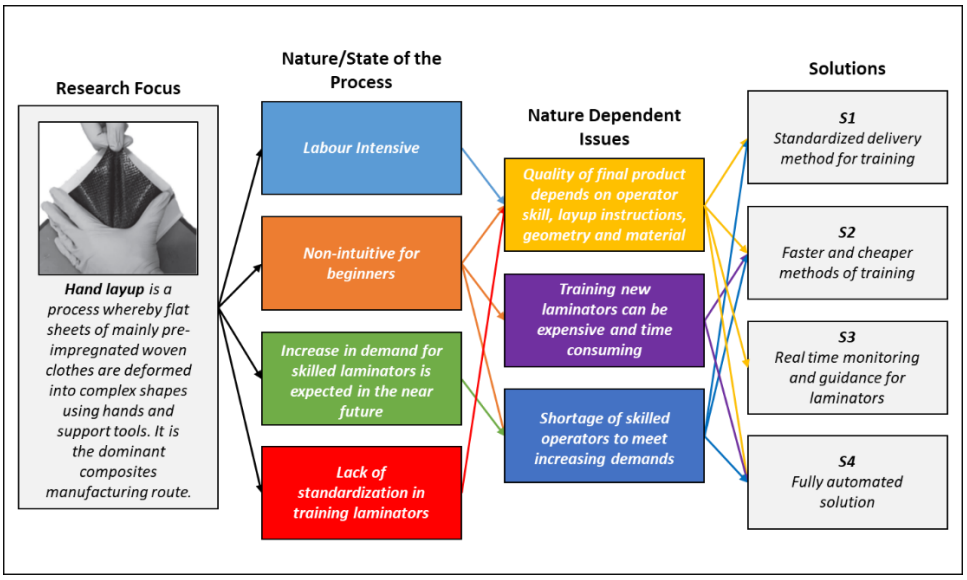


Figure 1.9 – Understanding research motivations

With this understanding, the research question of this thesis can be expressed as:

"Given a completely new mould geometry, how can we provide standardized training to multiple laminators around the world in a fast and economical manner, so that the final layups are identical to each other?"

As it will be evident in this thesis, through extensive research into multiple domains, including composites manufacture and computer science, the author has suggested and explored a potential solution path to the above research question that combines multiple technologies. This is better illustrated in Figure 1.10.

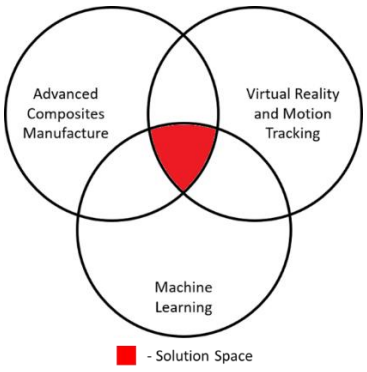


Figure 1.10 - Multidisciplinary solution path to the research question explored in this thesis

In the next chapter, a comprehensive review on how both Virtual Reality and Machine Learning can be used to solve the above research question is provided.

(Refer to Appendix: Table 1 for a summary of how each chapter of this thesis has presented the journey in pursuing these solutions.)

CHAPTER 2: Virtual Reality (VR) Training and Artificial Intelligence for Hand Layup

2.1 Introduction

In this chapter, the concept of using Virtual Reality technology for skills training within hand layup and how state of the art artificial intelligence techniques can be used to create such VR simulations will be explored. Firstly, the current state of skills and training around hand layup will be explored. This is followed by an introduction to VR technology and artificial intelligence. Review work related to VR presented in this chapter has been adapted from previous work carried out by the author in [56].

2.2 Current State of Skills and Training around Hand Layup of Prepreg

Although there exist many barriers in bridging the gap between technology and skills in the advanced composites industry, a potential workforce shortage, with the necessary skills to undertake the layup tasks, has been identified as one of the major concerns in the near term. Aligned to this is an existing and immediate need for re-skilling and/or up-skilling of the current workforce, as well as standardization of training delivery [57]. Some countries, such as the UK have stated aims of doubling its composite workforce within the near future [58]; although how such an increase in new workers (most likely from a variety of experiences and capabilities) can be quickly and efficiently trained remains unclear. A near chronic lack of access to resources and competent training (including evaluation processes) appears to have been a major issue to date, whether in the UK or in the off-shoring of parts overseas, leading to extended training schedules, excessive costs, owing to steep on-the-job learning curves and limited knowledge base capture/exploitation [57]. These risks are further heightened by the possible escalation of composite use into other sectors, and in increased product demand for those current sectors employing the material. Certainly, a mantra of ‘Bigger, Faster, and Cheaper’ has been adopted.

As established in the previous chapter, composites manufacturing is a complex and multi-stage process, with the layup stage requiring the most tactile skill and given the least in-process guidance. With such a heavy emphasis placed on the operator's skill and judgement the process by which these skills are acquired (initially, training and later, experience/practice), is vital. However [59] and [60] shows that employers are struggling to attract and keep skilled staff, and provides several references to this effect, in addition to their own case studies. In [60], organisations appear to single out training provided by further education colleges for composites. [60] also mentions that many companies are using in-house training to make up for the lack of training in the external labour market.

2.2.1 In-house Training - Shadowing

The use of in-house training is common in composites manufacturing. In interviews carried out in [61], all interviewees had received some form of in-house or on-the-job training. For more experienced laminators the trend has been to only have in-house training. This would take the form of shadowing. As a new starter, a laminator would watch and follow a more experienced laminator making a component, then making it themselves several times (one reported five), under supervision until the more experienced laminator felt they were sufficiently competent. Anecdotally, this has been the go-to method of training in composites lamination, one-to-one or small groups, making a component, hands-on, with supervision gradually reducing as the trainer feels confident in their trainee's ability.

This style of learning is not necessarily a bad method of teaching skills. However, it is also being carried out with no set curriculum. There may not be a formal review of the training process and there would be no formal reward or recognition. This means that, from an employer perspective, length of service is the only usable metric for determining skill level. If the laminators are being trained to make a particular component, it is not clear how much of that learning would be transferred to another component. It may even be necessary to re-train the laminator to fabricate a different component. This leads to a situation where two laminators with the same "skill level",

determined by their respective lengths of employment, having entirely different experiences and skill sets. A second issue with this type of in-house training is that it can enforce bad practice. If the only training and practice is in-house, then any deficiencies in technique will remain and the same bad practice will be passed down to the next generation or transferred to another company.

There is also a productivity argument against in-house training. Since it uses an experienced technician or laminator to carry out small-group training, it takes them away from other productive tasks that they could be doing. However, there is an argument in [61] that suggests that training others may be a good way to retain skilled personnel, by simultaneously trying to maintain their skills by passing them on and making an experienced employee, who may have considered leaving, invested in the future of their trainee.

There are some benefits to in-house training. The main benefit is that it can help to "indoctrinate" new starters into the practices employed in the company or at that location. Some form of in-house training or orientation is nearly always necessary. The second benefit is that the training received is certain to be relevant to the tasks the laminator will be doing. If the factory only makes one component from one material using one process, then in-house training will ensure they are not over-trained. This also has the benefit to the employer of not "training to leave". From conversations with managers in composites manufacturers, "training to leave" is a barrier to investing in training for new staff. By carrying out in-house training, which may only be valid in a single setting, the fear of "training to leave" can be alleviated.

2.2.2 External Training Courses

In addition to the usual practice of in-house training, there has been a recent expansion in the number of external companies and colleges which provide composites training. Composites UK, the composites trade association, lists 20 training providers on their website [62]. There is an array of learning materials, techniques and methods taught during these courses and an accompanying variation in length and cost. Some of the courses mentioned in [62] focus on aspects of composites beyond manufacture, some

focussing, or including at the least, design and many others dealing with in-service repair. Using [above] as a list, less than half of the available courses deal with hands-on, composite lamination, with even fewer having it as a principal focus. Composites, as an industry, is widely varied and includes many materials and processes, including repair, so the variety of available training is entirely justified. However, these various composites courses may not be properly differentiated in the minds of employers.

2.2.3 Other Training Methods

Alongside the usual practice of in-house training there are also National Vocational Qualifications (NVQ), which deals specifically with composites. Several other NVQ courses also mention or in some way involve composites, typically as an alternative material for machining etc. There are both Level 2 and 3 NVQs for composite manufacturing provided by various organisations and according to varied, but similar, assessment criteria. Examples include [63] and [64]. These courses should teach the required skills for composites layup, and would form a foundation on which to teach other, complimentary skills such as resin infusion, assemblies, machining etc.

Also, launched in the summer of 2017 is the Trailblazer Apprenticeship in composites [65]. This is a 3-4-year scheme for students aged 16+. This is in response to the conclusions from [59], and part of a concerted effort, spearheaded by the National Composites Centre, in recent years to improve the quality and standards of composites skills training.

In conclusion, the delivery of laminator training is yet to be standardized [61], and mainly comes in the form of one-to-one tuition/coaching within the institution/company [59]. Specialist training providers appear to be increasing in number but suffer the same standardization issues (explored in [66]) and limited flexibility to offer immediate on-site training. If this present situation is unsustainable, then through the rising demands for composite products [67], this problem will only become more acute. Composite laminators have a choice [68] and it is this choice and the lack of standardization in training delivery that leads to them developing their own set of techniques, skills, and methods of layup. More adaptable and novel solutions must be found, and this PhD

project tries to address these issues by exploiting Virtual Reality (VR) technology and the concept of Gamification.

2.3 Introduction to Virtual Reality (VR)

Figure 2.1 – Use of virtual reality technology in healthcare for surgery training [69]

A Virtual Reality (VR) system is a natural interaction technology where users are immersed in a computer simulated three-dimensional environment, given the ability to roam in the virtual world and interact with objects within it in a seemingly real or physical manner [70]. VR systems have mainly been developed to serve as (i) immersive multimedia/entertainment interfaces, (ii) immersive 3D problem solving systems, and (iii) networked collaborative work environments. VR systems are enabled by a combination of technologies that range from visual and audio technologies to tactile, haptic and limbic technologies to brain and cognitive and game technologies. Specific application needs aside, necessity for VR arises as a result of (i) a rapid growth in complexity of engineering tasks, (ii) optimization need of product development measures such as time, cost, quality and impact and utilization of resources, and assets, and (iii) the continuing globalization and labour/knowledge division [71]. VR technology has evolved steadily throughout the past three decades and has been adapted into various industrial applications. These include: military training (flight/battlefield/vehicle simulation and medical training), education (virtual classrooms/instructors), healthcare

(surgery simulation, phobia treatment, robotic surgery and skills training), entertainment (video games and virtual museums/galleries/theme parks/theatres), engineering (used extensively in the automotive industry in 3D prototyping of new vehicle designs prior to test builds: e.g. JLR Virtual Reality Centre in the UK [72] and design and assembly simulation of state of the art aircrafts: e.g. Airbus [73]), air travel (Airbus is in the process of developing in flight virtual reality systems to reduce passenger stress levels [73]) and sports (athletic training and performance analysis in golf, athletics, skiing, cycling and etc.) [74]. VR applications are numerous and continue to grow with the technology advancement, which is stimulated mainly by the video gaming industry.

While for simplicity, the term 'VR' has been used in this thesis to refer to any situation in which a user is immersed in a computer simulated 3D environment, a more complex and precise definition for 'VR' can also exist, if it is considered in line with 'Augmented Realty (AR)' and under the top level 'Mixed Reality (MR)' technology definitions, where there exist a continuum from the real world to a fully virtual experience. A comprehensive study on this continuum can be found in [75].

2.3.1 Virtual Reality for Skills Training

Technical training/knowledge transfer is among one of VR's most prominent applications. VR training provides several advantages compared to traditional video/written based training. An investigation carried out in [76] has proven that VR training is highly motivating for its users compared to traditional training methods. This is because VR training encourages active participation rather than passivity. VR can illustrate certain features and processes with higher accuracy and detail by providing the user with complete freedom of viewing angles and proximity [77]. VR systems, when used for training, creates a platform in which users can repeat a set task as many times as needed at their own pace away from social pressure to obtain the required skills and knowledge prior to physically performing it. VR also enables simulation of high-risk situations in a controlled environment and training of large groups of people remotely at low cost.

Despite the advantages offered by VR, it is essential to understand when VR should be considered as a training tool and when it should not. VR should be considered as a training tool when (i) real training is of high risk, (ii) VR training provides a more interactive experience than real training, (iii) access to resources and infrastructure required for real training are limited, (iv) telepresence of trainees from different geographical locations are required for group training and (v) mistakes made by the trainee during training could be demoralizing to the trainee, harmful to the environment, capable of causing damage to equipment and property, or costly. VR shouldn't be used as a training tool if, (i) cost of using VR training is too high to justify, considering the expected learning outcome, (ii) interaction with a real trainer is essential, (iii) use of VR is physically or emotionally damaging, (iv) use of VR causes "literalisation" (a simulation so convincing that causes users to confuse the model with reality) and (v) use of VR is impractical (VR training may not be suitable for activities involving fine details. This is due to limitations in resolution of motion tracking hardware. However, this issue is being addressed by the rapid growth in motion tracking technologies) [77].

With this understanding, it can be explained why VR training is suitable for hand layup of composite materials, especially for prepreg material. Layup of prepreg is carried out in a temperature/humidity-controlled environment. Prepreg material requires cold storage to prevent complete cure and hence must be defrosted prior to layup. Prepreg material is relatively expensive compared to the use of dry fibre and resin. Therefore, layup of prepreg is of high cost and material and infrastructure are not readily available. Add to this the fact that the layup procedure is complex and hence requires thorough repetitive training, which is usually provided by an expert laminator who will have to be taken out of manufacture for the training period; VR then becomes a compelling platform for training composite laminators.

2.3.2 Technology Review: VR

Figure 2.2 shows a simplified diagram of subparts that make up a typical VR system and how they interact with each other. The computing section, which controls the complete

VR system, is the device to which tracking, control, and audio devices are connected and at the same time perform calculations related to user movements, object handling and graphic processing. Finally, this unit provides a RGB signal to the projection device.

Tracking of user motion and control devices are the two modes of inputs that serve as main input mechanisms in a VR system. Tracking can be optical, magnetic, ultra-sound, mechanical or inertial; each of them having their advantages and disadvantages subject to the application. Common control devices include joystick, trackball, mouse, keyboard, or wireless pads.

The virtual environment is presented to the user using a screen. Screen types range from simple desktop monitors to head mounted displays to single/multi-wall curved screens to front/back projected screens. Each of these screen types offers different levels of immersion to the user.

Audio unit is optional and ranges from simple stereo headsets to 3D surround sound depending on the application requirements.

The haptic unit is also optional, and it provides force feedback to the user upon interaction with virtual objects. The haptic system varies from a simple data glove providing touch sensation through electrical pulses, to full exoskeletons providing force feedback to the entire body [78].

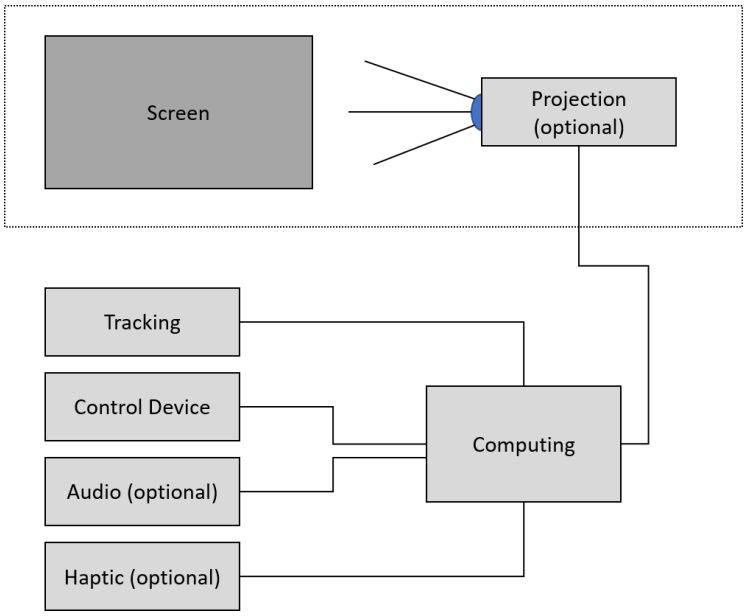


Figure 2.2 – Virtual reality system synthesis showing interactions between its subparts

2.3.3 Immersion in VR

The level of immersion is a key concept that drives the design of VR training systems. Level of immersion is a measure of the sense of presence created in the user's mind. A 3D application on a desktop screen where a keyboard and a mouse are used as input devices can be categorized as low immersion, while use of head mounted displays with stereoscopic 3D displays and head tracking, 3D body tracking, and real time haptic feedback can be categorized as high immersion. Accuracy and precision of sensors and feedback devices used also contribute to the level of immersion. Studies have shown that higher levels of immersion lead to increased levels of concentration of trainees, awareness of spatial relationships, paths and proximity and skill transfer [79]. Increase in the level of immersion is in general associated with an increase in cost of the VR system. The level of immersion required depends on the type of application VR is being used for. Therefore, it is important to find a balance between cost and immersion.

2.3.4 Head Mounted Displays (HMDs)

HMDs are discussed in detail as the VR systems presented in this thesis are all based on HMDs. HMDs enable the design of highly immersive VR systems at a moderate cost compared to some of the high cost VR setups such as CAVE Automatic Virtual

Environments which take the form of a cube like space in which multiple projectors display images on to walls to provide a fully immersive experience. As a micro display is worn close to the user's eyes, a compact device can cover the complete field of view. Although, a number of semi-transparent HMDs that allow simultaneous viewing of the real world and computer-generated virtual worlds are available, the majority of HMDs are occlusive. By feeding the right and left eye with separate images, stereoscopic 3D viewing is possible, which allows the user to perceive depth. When using occlusive HMDs, low-latency tracking of head movements becomes essential to adapt the displayed image to the orientation of the user's head to avoid motion sickness. Although it is easy to immerse the user in to his/her own virtual world, co-operation is complicated due to non-transparent displays. Avatars or video-captured images can be used to overcome this issue [78].

HMD based VR systems have existed for decades. The promise of VR has always been enormous. However, until recently the VR industry has failed to deliver this promise. An industry that almost shut down completely in the mid 90's was brought back to life recently by the HMD VR headset called the "Oculus Rift". This was considered the most technologically advanced and robust HMD VR system available at the time the research presented in this thesis was carried out [80]. Since then many iterations of the Oculus Rift have been released and the latest version offers low latency head tracking using optical and inertial sensors and negligible blur and judder via the use of a low persistence organic light emitting diode (OLED) display. Detailed technical specifications can be found in [81]. This device is not fully self-contained. It must be tethered to an external computer. This limits mobility of its user and its portability as a device. However, Oculus has recently introduced "Oculus Go" [82], a fully self-contained VR device. HTC VIVE [83] and Samsung Odyssey [84] are few competitors of the Oculus Rift.

Rapid increase in Central Processing Unit (CPU) and Graphic Processing Unit (GPU) power has led to smartphones being considered as a self-contained HMD VR platform [85]. State of the art smartphones have an inbuilt inertial measuring unit (IMU) for head tracking, high resolution displays capable of displaying high quality and accurate

graphics, and wireless connectivity requirements to connect input devices. Add to this the facts that smartphone-based VR is completely wireless, enabling complete mobility to the user, a large percentage of people in developed countries own a smartphone and updates it in regular intervals, smartphones then become a compelling VR platform. However, current smartphones have not been designed for low latency VR, i.e. even though the screens are of high resolution, their pixel response time and refresh rates are not high enough to keep up with fast head movements, hence could cause issues related to motion sickness. Furthermore, mobile CPU and GPU are relatively limited in their processing power compared to desktop computers. That being said, use of VR as a platform for composite layup training does not require rapid head movements or excessive processing power. Therefore, a smartphone-based VR system could prove to be a better choice compared to an Oculus Rift for such applications with its added benefits such as relatively low cost, ease of access and mobility.

2.4 Understanding the Need for Artificial Intelligence in Designing Virtual Reality Simulations for Hand-Layup Training

Through 2 case studies carried out, as explained in Chapters 3 and 4, it was identified that one of the major bottlenecks in designing Virtual Reality training aids or any other form of manufacturing instruction set for hand laminators in laying up complex composite parts is the inability to predict the best order of physical manipulations required to form the reinforcement on to the tool. Laminators currently figure out this best sequence of actions on the go, while performing the layup, and there is no method to validate that the sequence of actions a laminator decides is optimal for a given mould shape. This type of approach also leads to standardization issues as different laminators come up with different ideal sequences of actions that they believe is the most suitable for a given mould shape. A possible solution to this issue is proposed in this chapter centred around Artificial Intelligence (AI) techniques. Such a solution would also enable the automatic generation of Virtual Reality training simulations for new mould shapes and thereby avoiding the tedious process of having to hand code each single step of VR training simulations. Few attempts have been made in the past to use classical drape

simulators to output an ideal sequence of actions for a given mould shape. Therefore, it is worth reviewing these past attempts before presenting the novel solution.

2.5 Drape Simulators to Inform Hand Layup of Complex Composite Parts

In the very early stages of the composite industry, it became clear that the manipulation of unidirectional prepreg by hands to form the reinforcement on to the tool was near impossible due to the material's tendency to split, tear, and wrinkle upon the application of manual forming loads. In addition, when it comes to parts with high double curvatures, use of unidirectional prepreg was certainly impractical due to their inability to provide any in plane shear deformation. The solution to this was the invention of woven prepreg. Woven reinforcement can be easily formed on to complex shapes, as they offer significant amount of in plane shear deformation and are less sensitive to operator handling skills compared to unidirectional prepreg [53].

Fairly soon after the introduction of woven prepreg, significant amount of work was put in to creating computer simulations capable of mimicking the behaviour of such materials. Computer models with increasing complexity have been developed since the 1970s [43], [86], [87], [88], [89]. It is certainly worthwhile reviewing the current state of the art of drape simulators prior to discussing their use for informing hand lamination.

2.5.1 Drape Simulators

Drape simulators attempt to simulate the behaviour of woven clothes during forming. These simulators focus on modelling the deformations of plies in the form of in plane shear and out of plane deflections and residual stresses. Modelling final fibre orientations is their primary focus as it is a key input to Finite element modelling (FEM) packages. Current drape simulators are capable of predicting final fibre/tow directions, in plane shear distribution and out of plane deflection of the final drape pattern of woven clothes. Drape simulators are also capable of predicting whether a certain mould shape can be draped when a particular type of woven reinforcement is used. But, what they can't provide is step by step instructions on how to achieve the final predicted drape pattern.

There are two main categories of drape simulators: FEM schemes and Kinematic modelling schemes. Each of these schemes are reviewed below.

2.5.1.1 Finite Element Modelling Schemes

Most of the commercially available drape simulators such as Aniform, PlySim, CATIA Composites Fibre Modeller, and FibreSim tend to use the FEM scheme. FEM schemes are usually based on energy balance or force equilibrium. FEM schemes tend to be more computationally expensive compared to the kinematic modelling schemes. This is one major reason why Kinematic modelling schemes tend to be used more often. FEM based drape simulators opt for several approaches such as discrete [90], semi-discrete [91], and mesoscopic [92] due to the multiscale nature of composite materials [93].

2.5.1.2 Kinematic Modelling Schemes

In contrast with FEM schemes, kinematic modelling schemes use low computational resources by using a less accurate representation of material properties and processing conditions during draping [93]. The simulations usually start off with a predefined component surface geometry, and in most cases assumes an initially orthogonal woven net. An initial point of contact is set by the user and this step is followed by defining an initial warp and weft tow path over the surface starting from the initial point of contact. The woven reinforcement is modelled as a net made of cells with 4 sides that have equal lengths. Finally, the surface is covered using this net in an iterative process in such a way to ensure contact between the cell edges and the mould surface [53]. Virtual Fabric Placement (VFP) [51] is an example for a drape simulator using the kinematic modelling scheme. VFP is unique in that it allows the user to define where the net is first stuck down [16]. VFP has shown that for the same mould shape, multiple shear deformation patterns can be achieved based on where the net is first stuck down. This is an important finding in that time to layup is significantly affected by the shear distribution in the woven net.

2.5.2 Informing Hand Lamination Using Current Drape Simulators

Few attempts have been made to reverse engineer the output from current drape simulators to construct layup instructions. In [22], M. Elkington proposed a system named Rectangular Indicator Field (RIF) using outputs from VFP, where a pattern of rectangles was used on the prepreg to communicate to the laminator the required shear deformation that must be generated on woven prepreg to complete a layup. Figure 2.3 shows how this system uses rectangles to inform the laminator on shear deformation. In [53], S.G. Hancock introduced a novel strategy for generating unambiguous manufacturing instructions for laminators via enhancing the output from VFP. This particular strategy relates experimentally determined layup rules to the designed tow alignment pattern and uses tow curvature to extract the dynamic behaviour of the forming process. The prepreg is divided in to sections of similar in plane curvature and an order of forming is defined for these sections. Figure 2.4 illustrates an example of this system. While these methods provide a certain level of guidance to laminators, it has been concluded that the predicted fibre orientations from a kinematic forming simulation does not provide all the information required for shop-floor operatives to match those orientations [53].

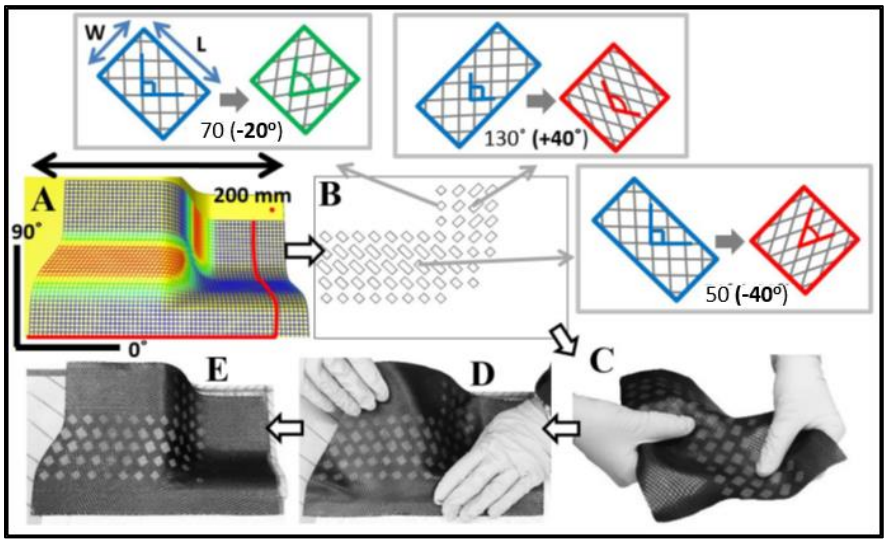


Figure 2.3 - Lamination and use of Rectangular Indicator Field (RIF); (A) Virtual Fabric Placement diagram; (B) RIF Pattern with schematics of 3 example rectangles; (C) Ply marked with RIF pattern being presheared; (D) Presheared ply being fitted to the tool; (E) Finished ply. (Images not to scale) [23]

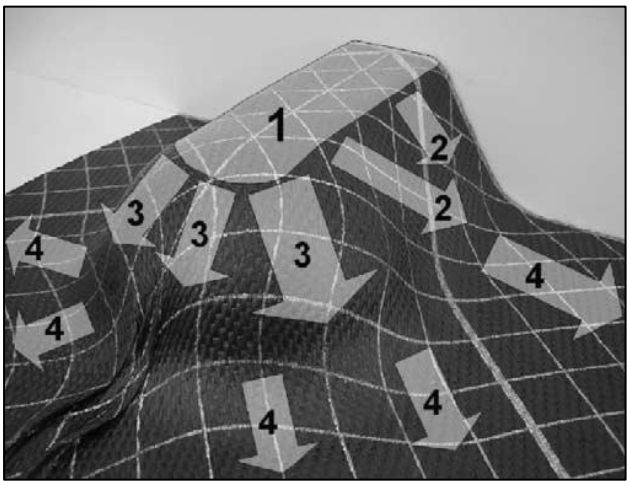


Figure 2.4 - Order of drape for intuitive approach [53]

2.6 Rethinking the Fundamental Approach of Drape Simulators

The inability of drape simulators to produce step-by-step instructions required to form woven reinforcement to shape even after decades worth of drape simulator development suggest that perhaps we must question the fundamental approach taken by them. All current drape simulators have been designed with the primary aim of predicting the final drape pattern in mind. Predicting the sequence of actions required

to achieve the final drape pattern to date has been a secondary requirement despite its importance in composite manufacture.

It has already been established that the predicted fibre orientations from drape simulators are insufficient to generate a sequence of physical manipulations required to form woven reinforcement in to shape. This suggests that there is a missing fundamental dimension in current drape simulators. The solution to this issue is perhaps not that far out of reach. In fact, the author believes that the solution to this issue is embedded in the manual process itself. The author believes that all current drape simulators have failed to model one fundamental aspect of the manual layup process. That is the human learning process, or in other words current drape simulators have failed to model the process that occur within the brain of a hand laminator while the laminator is trying to figure out how to layup a new mould shape. How laminators figure out their personal best sequence of physical manipulations to layup is through a fundamental animal learning process called reinforcement learning. Simply put, reinforcement learning is learning to achieve a goal through trial and error. A more detailed definition of reinforcement learning is provided in a subsequent section of this chapter. Modelling the reinforcement learning process is only one part of the solution. The other change that needs to be made to drape simulators in order to be able to use reinforcement learning is to replace the modelling of final deformation of reinforcement with progressive reinforcement deformation. I.e. we need to be able to model the deformation of the reinforcement and its interaction state with the mould upon every single physical manipulation that is made to the reinforcement.

2.7 Reinforcement Learning: An Introduction

The first thing that comes to mind when thinking about the nature of learning is that humans learn by interacting with their environment. When an infant is playing, waving its arms or looking about, it does not have an explicit teacher, but it has a direct sensorimotor connection with its environment. Exploiting this connection gives the infant a wealth of information about cause and effect and what to do to achieve goals. Throughout our lives, we all exploit this feedback mechanism to gather knowledge about

our environment and ourselves. No matter what goal we are trying to achieve such as learning to drive a car or to play a sport, we are always aware of how our environment responds to our actions, and we try to influence what happens through our actions. This type of learning from interaction is a fundamental idea that underlies all theories of learning and intelligence [94].

In trying to solve the major limitation of current drape simulators as previously mentioned in this chapter, what we require is a computational approach to learning from interaction. In other words, what we require is a computational reinforcement learning approach. In the computational world, reinforcement learning is defined as "learning what to do – how to map situations to actions – so as to maximize a numerical reward" [94].

2.8 Elements of a Reinforcement Learning System

Before understanding reinforcement-learning algorithms in more depth, it is important to understand primary elements of a reinforcement learning system. In order to understand the elements of a reinforcement learning system, a very simple video game named "Breakout" by Atari from the 1970's will be used. Refer to Figure 2.5 for its game components.

The rules of this game are very simple. The user is given control of the paddle, which can be moved in the left or right directions. The user is supposed to break all the bricks in the current level by bouncing the ball off the paddle without letting it go pass the paddle to advance to the next level.

2.8.1 Reinforcement-Learning Elements: Goal(s)

Every reinforcement learning system must contain at least one goal. In the case of computational reinforcement learning systems, this goal is always to either maximize or minimize a certain numerical value. In the case of Atari's Breakout game, the goal is to maximize the game score by breaking as many bricks as possible while minimizing the loss of lives.

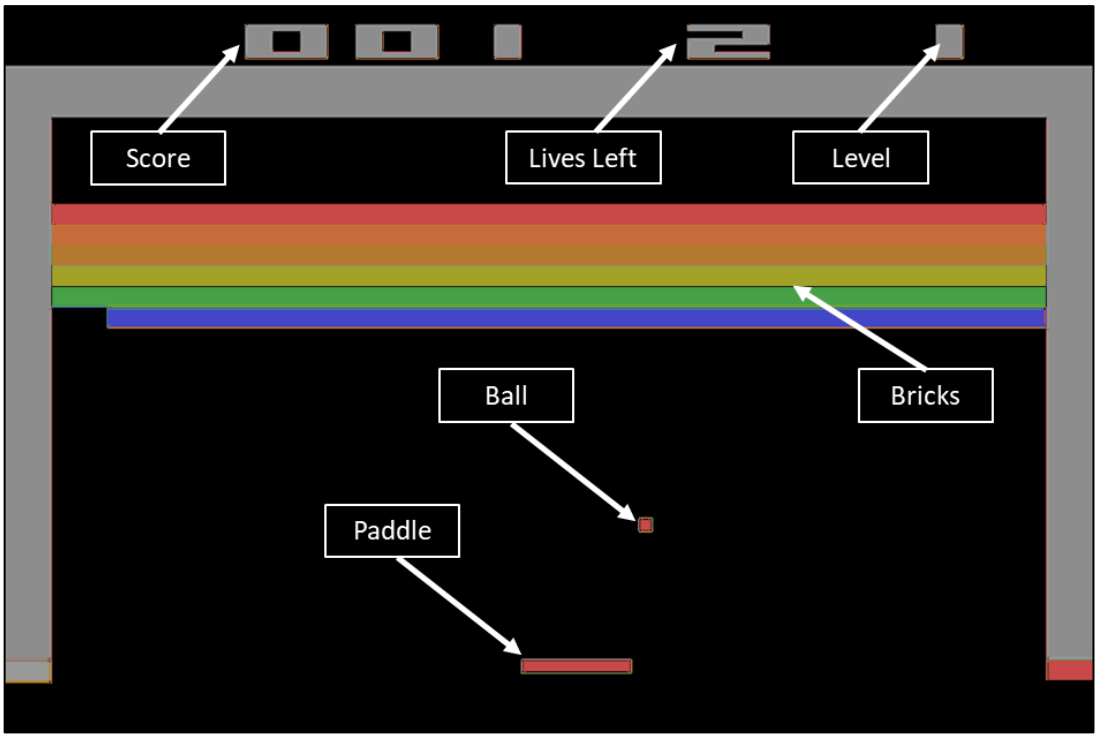


Figure 2.5 - Atari's Breakout: Game componenets

2.8.2 Reinforcement-Learning Elements: Agent

The agent is the learner or the decision maker in a reinforcement learning system. If a human were playing Atari's Breakout video game, the human would become the agent. However, non-human digital agents can also be designed. Digital agents play a significant role in the design of artificially intelligent drape simulators and hence will be discussed in more depth in a subsequent section of this chapter.

2.8.3 Reinforcement-Learning Elements: Closed Environment

The thing the agent interact with or everything outside of the agent is defined as the environment of a reinforcement learning system. The environment has a finite volume/area and hence is called a closed environment. If a human or a digital agent were playing Atari's Breakout video game, every object on the game screen would belong to the closed environment.

2.8.4 Reinforcement-Learning Elements: Actions

The agent is given the ability to take actions upon the environment that changes its state. In the case of Atari's Breakout game, the agent has the freedom to take three distinct actions at any particular time step: 1. Move the paddle to the right, 2. Move the paddle to the left, and 3. Keep the paddle still.

2.8.5 Reinforcement-Learning Elements: Rewards

The agent receives numerical rewards based on the actions it takes as a feedback mechanism. There are three types of instantaneous rewards that the agent can receive: 1. Positive reward, 2. Negative reward, and 3. Zero reward.

2.8.6 Reinforcement Learning Elements: State(s)

A state is an instantaneous representation of the environment that is been presented to the agent. In the case of Breakout, an instantaneous state can be defined as the position, colour and size of every single game object. At a more fundamental level, a state can also be defined as the RGB values of every single pixel on the screen.

2.9 Reinforcement Learning: Broadening the Understanding

It is important to understand the distinction between reinforcement learning and supervised learning, the type of learning that is studied most in the machine learning research field, statistical pattern recognition and artificial neural networks. Supervised learning is learning from an already available data set collected from a knowledgeable supervisor. While this is an important form of learning, supervised learning alone is insufficient to learn through interaction. In addition, in interactive situations, it is impractical to obtain a training data set that are both correct and representative of every situation the agent has to act in. In a previously unseen situation, where learning would be most beneficial, an agent must have the capability to learn through its own experience [84].

2.9.1 Sub Elements of a Reinforcement Learning System

In addition to the main elements defined in Section 2.8, there are 3 main sub elements of a reinforcement learning system that are crucial in understanding the reinforcement learning process [94].

2.9.1.1 The Policy

The policy of the agent is a description of its behaviour at a given time. In other words, the policy is a mapping between a given state and the actions to be taken at that state. The form of the policy can vary from a simple function or a look up table to a computationally expensive artificial neural network. The policy itself is sufficient to describe the behaviour of the learning agent and hence it is its core. In most cases, the policy is calculated statistically, because, it may not be predicted precisely.

2.9.1.2 The Reward Function

The reward function maps each state-action pair of the environment to a numerical reward (r). The reward determined by the reward function for each state-action pair indicates the desirability of taking that particular action at that particular state. A higher reward represents a higher desirability and vice versa. A true reward function acts as the basis for altering the policy of the agent. For example, if an action taken by the agent following the policy yields an unfavourable reward, then the policy can be changed such that a different action is to be explored in the future when the agent is presented with that same state.

2.9.1.3 The Value Function

While, the reward function is an indication of what action is the best to take to yield the highest instantaneous reward at a given state, the value function indicates which action is the best to take to maximize the total future reward. In other words, the value (q) of a particular state-action pair is the total reward the agent is expected to accumulate over the future if that particular action was taken in that state. With the presence of a value function, the agent may be forced to take actions that yield unfavourable rewards

in the short term, which will result in a larger future total reward. Values are considered secondary to rewards. This is because, without rewards, values do not exist, and the purpose of estimating values is to maximize rewards. However, values are more important to the agent when making decisions.

2.9.2 The Balance between Exploitation and Exploration

The balance between exploration and exploitation is a major challenge that arises in reinforcement learning. In order to obtain a large reward, the agent must prefer to take actions that it had already taken in the past and had proven to be favourable. However, the discovery of such effective actions is not possible without taking actions that it had never taken before. In other words, the agent must exploit actions that it already knows yields favourable rewards, and at the same time, the agent must continue to explore actions previously not taken in order to find more favourable actions than the currently most favourable ones. The dilemma here is that the agent cannot succeed at achieving its goal if either exploration or exploitation is pursued exclusively. The agent must find the right balance between these two methodologies in order to succeed in achieving its goal. We will now explore few methods available for action selection. Few notations are introduced at this stage to make exploration of action selection methods easier.

Q_t (a) = Estimated value of action 'a' at time step 't'.

A_t^* = Action with the highest estimated value at time step 't'.

2.9.2.1 The Greedy Method

The greedy method is the simplest action selection method and it always selects the action with the highest estimated value (A_t^*) at every time step 't'. This is a full exploitation method and is very unlikely to allow the agent to achieve its goal as explained previously.

2.9.2.2 The ε-Greedy Method

An alternative to the greedy method is to select the action with the highest estimated value most of the time, but every once in a while, for example with probability ε, select

an action with equal probability among all other actions except the one with the highest expected value.

2.9.2.3 SOFTMAX Action selection

While the ε-Greedy method offers some balance between exploitation and exploration in reinforcement learning, its major drawback is that when it explores it chooses with equal probability among all other actions. This means there is an equal chance to the agent picking the worst possible action and the action that is second best. In reinforcement learning tasks, where the worst action is highly undesirable, this is unsatisfactory. The solution to this issue is to vary the probability with which actions are picked as a graded function of action values. This type of action selection is called SOFTMAX action selection and the most commonly used SOFTMAX action selection criteria is named Gibbs or Boltzmann and is represented as,

$$Probability(a) = \frac{e^{Q_t(a)}}{\sum_{i=1}^{n} e^{Q}} \qquad \textbf{Equation 2.1}$$

In Equation 2.1, 'e' is the exponent of natural logarithm and 'n' is the total number of actions available to the agent at a given state.

2.9.3 The Agent-Environment Interface

The agent interacts with the environment continually. The agent selects actions, and the environment responds by presenting the agent with a new state based on the effect the action had on the environment. The environment also provides numerical rewards to the agent, which the agent tries to maximize over time. More specifically, the agent interacts with the environment at each of a sequence of time steps, t = 0, 1, 2, 3, 4…T. The agent is presented with an instantaneous representation of the environment's state, $S_t \in S$, where S is the set of all possible states available to the environment. Based on S_t, the agent takes an action $A_t \in A(S_t)$, where $A(S_t)$ is the set of actions available at state S_t. After one time step, as a result of action A_t, the agent receives a numerical

reward R_{t+1}, and is presented with state S_{t+1}. Figure 2.6 represents the interaction procedure between the agent and the environment in a reinforcement learning system.

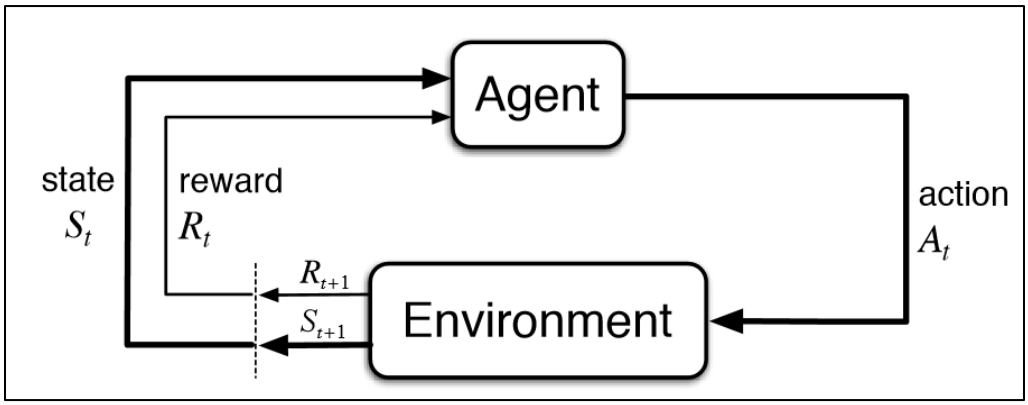


Figure 2.6 - The agent-environment interaction in reinforcement learning

At every time step, the agent utilizes a mapping from states to probabilities of taking each possible action available at that state. This mapping was previously name as the agent's policy and is defined as π_t, where $\pi_t(a|s)$ is the probability of taking action 'a' given state S_t. The objective of a reinforcement learning agent is to optimize this policy function through interaction with the environment such that the cumulative reward in the long run is maximized.

2.9.4 Formalizing the Future Cumulative Reward: Returns

It was mentioned previously that the aim of the agent is to maximize the future cumulative reward in a reinforcement-learning environment. In this section, this idea will be formalized. If we represent the series of rewards received by the agent after time step t as R_{t+1}, R_{t+2}, R_{t+3}, R_{t+4} ……, then what we seek to maximize is the future expected return denoted by G_t, where,

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... R_T. \qquad \textbf{Equation 2.2}$$

In Equation 2.2, 'T' represents the final time step. This approach works in situations where the interaction between the agent and the environment naturally breaks in to sub sequences, called episodes, such as plays of a game or any kind of repetitive

interactions. The final state of these repetitive episodes is called the terminal state, which is then followed by resetting the environment to its starting state. Tasks with this kind of episodes are called episodic tasks. However, there are cases where the agent-environment interaction does not naturally break in to episodes, instead goes on continually without an end. These are called continuing tasks. For the purpose of work presented in this thesis, only episodic tasks are relevant. Therefore, continuing tasks are not discussed furthermore. The additional concept required in formalizing the return for an episodic task is that of discounting. With the addition of this concept, the agent's goal is to maximize the sum of the discounted rewards it receives in the future. In other words, the agent chooses action A_t to maximize the expected discounted return:

$$G_t = R_{t+1} + \Upsilon R_{t+2} + \Upsilon^2 R_{t+3} + ... \Upsilon^{(T-t-1)} R_T = \sum_{k=0}^{T-t-1} \Upsilon^k R_{t+k+1}. \qquad \textbf{Equation 2.3}$$

In Equation 2.3, 'ϒ' is called the discount rate and $0 \leq \Upsilon \leq 1$. The discount rate determines the present value of future rewards. A reward received 'k' time steps in the future is perceived as being worth only Υ^{k-1} times its reward value if it was received immediately.

2.9.5 The Markov Property

In a reinforcement learning system, the agent makes decisions based on the state it has been presented with. In this section of the chapter, we will be discussing about the type of information we should and should not expect a state to provide. The state signal in general is not expected to contain all information about the environment, or even all information required to make a decision. For example, if the agent is a paramedic that arrives at a road accident, we cannot expect the paramedic to instantaneously know everything about the internal injures of the patient. In many reinforcement-learning systems, there will be hidden state information in the environment, and this hidden information would be useful to the agent if it knew them, but the agent cannot yet know them because it has not received the relevant sensory information from the environment. Therefore, we cannot judge the agent as faulty for not knowing everything about the environment, but only for forgetting information that it already knew.

With this understanding, what we require is a state signal that retains sufficient past information in a compact manner in such a way that all relevant information is retained. This often requires the state signal to contain information beyond the immediate sensory outputs from the environment, but never more than the entire history of that particular episode. Such a state that manages to retain sufficient information is said to be Markov, or to contain the Markov property. An example of a Markov state would be the current state of a checkerboard that contains information on all the pieces present on the checkerboard. Such a state would contain sufficient information about the sequence of actions that led to the current state. Despite such a state not containing much of the information on the sequence of actions that led to the current state, it contains all information required to make any further future decisions. Another example is the current position and velocity of a cannon ball. In order to predict the future position or velocity of the cannon ball we would not require information on how the cannon ball achieved its current position and velocity.

In this section, a mathematical representation of the Markov property will be provided. For this purpose, a reinforcement learning system with a finite number of states and reward values will be considered. Consider how the system responds at time 't+1' to an action taken at time 't'. In the most fundamental case, the behaviour of the system can only depend on everything that happened at time 't' and before that. In such a case, the response of the system at time 't+1' can be defined as:

$$\mathbf{Pr}\ \{R_{t+1} = r_t, S_{t+1} = s' \mid S_0, A_0, R_1, \ldots., S_{t-1}, A_{t-1}, R_t, S_t, A_t.\} \qquad \textbf{Equation 2.4}$$

Where, 'S' = set of all possible states, s' = next state and r_t = reward received at time strep 't'. For the reinforcement learning system described in Equation 2.4, if the state signal has the Markov property, then the system description can be reduced to:

$$\mathbf{Pr}\{R_{t+1} = r_t, S_{t+1} = s' \mid S_t, A_t.\} \quad \textbf{Equation 2.5}$$

Therefore, a state signal is only considered to have the Markov property if and only if Equation 2.4 is equal to Equation 2.5.

2.9.6 Markov Decision Process

In this section, an example provided in [94] will be used to explain a reinforcement-learning task that satisfies the Markov property. Such a task is named a Markov Decision Process (MDP). For this example, a finite MDP is used where the state and action space are both finite. 90% of the modern reinforcement learning process can be understood using finite MDPs.

For a finite MDP, given any state s and action 'a', the probability of each possible next state, s', is:

$$p(s' \mid s, a) = \mathbf{Pr}\{S_{t+1} = s' \mid S_t = s, A_t = a\}. \quad \textbf{Equation 2.6}$$

Such quantities are named transitional probabilities. Similarly, we can define the expected value of the next reward given the current state 's', and current action 'a', together with any next state s', as:

$$r(s, a, s') = \mathbf{E}[\, R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s' \,]. \quad \textbf{Equation 2.7}$$

Example - Recycling Robot MDP:

The goal of the agent (robot) is to collect empty soda cans in an office environment. The agent has sensors to detect cans. It also has arms and grippers capable of picking up cans and placing them in an on-board bin. It is powered by a rechargeable battery. The agent has 3 possible actions: (1) search for cans for a period of time, (2) remain

stationary, (3) head back to home base to recharge its battery. The decision to take a certain action must be made periodically or after a certain event happens, such as finding an empty can. The best action to take to find cans is obviously to actively search for them, but this action runs down its battery. However, staying stationary does not deplete its battery charge. Whenever it is actively searching for cans there is a risk of completely depleting its battery's charge. This will result in a shutdown of the robot, and someone will have to rescue the robot from this state and this scenario results in a large negative reward. The robot can distinguish between two levels of battery: (1) high and (2) low. So the state of the robot is S = {high, low}. Let us name the robot's actions as: (1) wait, (2) search and (3) recharge. Recharging when the battery charge state is high is unnecessary, so this is not included in the action set for this state. Then the robots action sets become:

A (high) = {search, wait}

A (low) = {search, wait, recharge}.

An active search can always be performed without the risk of completely depleting the battery when the battery charge level is high. An action of active search with an energy level of high leaves the battery at a high energy level with probability α and reduces charge level to low with probability $(1-\alpha)$. An active search performed with the battery level low leaves the battery level at low with probability β, and results in a complete depletion with probability $(1-\beta)$. In the latter case, as previously mentioned, the robot must be rescued for a recharge. Each can collected by the robot yields a positive unit result and each rescue yields a -3 reward. Let r_{search} and r_{wait} with $r_{search} > r_{wait}$ denote respectively the expected rewards from searching and waiting. In addition, no cans can be collected while going back home for a recharge or while the battery is completely depleted. With this definition, this system becomes a finite MDP, and transition probabilities and expected rewards of this system can be illustrated by a transitional graph as shown in Figure 2.7.

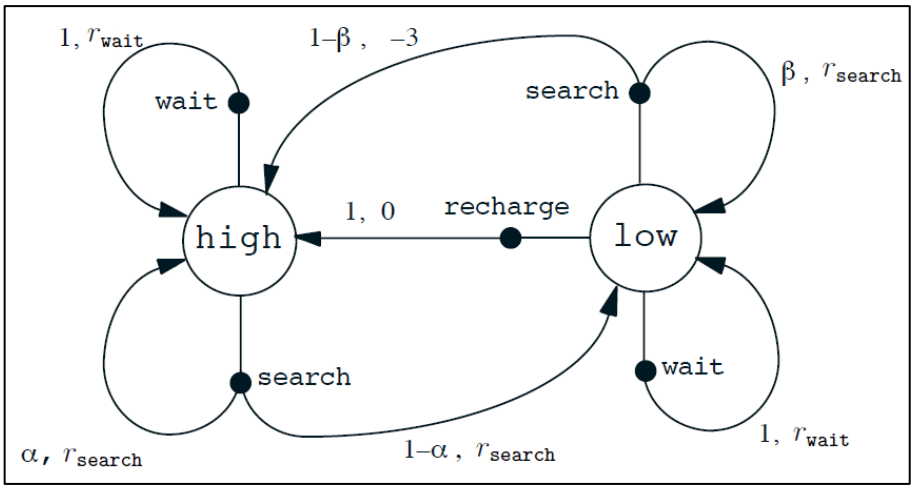


Figure 2.7 - Transition graph for the recycling robot

In this graph, there are two types of nodes: (1) state nodes and (2) action nodes. Each possible state is represented by a large open circle with the label of that state inside of it. A small solid circle is used to represent each action-state pair. Starting from a particular state 's' and taking an action 'a' moves you along the line from state node s to state action node (s, a). The environment then responds by moving you to the next state node via one of the arrows leaving from the state action node (s, a).

2.9.7 Value Functions

All reinforcement systems involve calculating a value function. A value function for a particular reinforcement system is a function of states (or state-action pairs) that estimate how desirable it is for the agent to be in a particular state (or how desirable it is to take a particular action given a particular state). The notion of "how desirable" in this case is represented by the expected reward at a particular state (or particular state action pair). The total expected future return for an agent of course depends on the actions it takes. Therefore, value functions are defined with respect to policies [94].

To avoid any further confusions, it is important to understand the relationship between the return (G_t) and the value function. A value function does not equal the 'return function'. Instead, a value function calculates the expected return from being in a certain state, or taking a specific action in a specific state, whereas the return is a measured

value that represents the actual discounted sum of rewards received following a specific state or state/action pair.

It was previously defined that a policy π (a | s) is the probability of taking action 'a' given state 's'. Following this definition, the value of state 's' under policy π, denoted by $v_\pi(s)$, is the expected return if policy π was followed starting at state 's'. For Markov Decision Processes, $v_\pi(s)$ can be formally defined as:

$$v_\pi(s) = \mathbf{E}_\pi [G_t | S_t=s] = \mathbf{E}_\pi[\sum_{k=0}^{T-t-1} \Upsilon^k R_{t+k+1} | S_t = s]. \quad \textbf{Equation 2.8}$$

In Equation 2.8, $\mathbf{E}_\pi$ [.] refers to the expected value if the agent follows policy π, and 't' is any time step. Function v_π is named the state-value function for policy π.

On the other hand we can define $q_\pi(s, a)$, the value of choosing action 'a' in state 's' under policy π. In other words, this is the expected return from starting at state 's' and taking action 'a' following policy π. This can be formally defined as:

$$q_\pi(s, a) = \mathbf{E}_\pi [G_t | S_t=s, A_t = a] = \mathbf{E}_\pi[\sum_{k=0}^{T-t-1} \Upsilon^k R_{t+k+1} | S_t = s, A_t = a]. \quad \textbf{Equation 2.9}$$

q_π is named action-value function for policy π. The value functions v_π and q_π can be estimated from experience. For example, if an agent follows policy q_π and maintains an average, for each state encountered, of the actual returns that have followed that state, then the average will converge to the state's value, v_π (s), as the number of times that state is encountered approaches infinity. If separate averages are kept for each action taken in each state, then these averages will similarly converge to the action values, q_π (s, a). For more clarity on this concept, refer to Sections 2.2 and 3.5 of [95]. This kind of approximation is called Monte Carlo approximation as they involve averaging of many random samples of real returns. This type of a method would be well suited for the example of the recycling robot shown in section 2.9.6 where there is only a handful of states and actions. For reinforcement learning tasks with a large number of states and

state-action pairs, keeping averages for all of them becomes impractical and it might take a significant amount of time for the values to converge to accurate values. A more generalized approach is required in such situations and this issue will be discussed in detail in a subsequent section of this chapter.

2.9.8 Monte Carlo Methods

Monte Carlo methods are used in reinforcement learning when the agent has to learn to thrive in an environment without complete knowledge of the environment. In the example of the recycling robot in section 2.9.6, the agent had all information about the environment at the beginning. However, this is not the case in many reinforcement-learning tasks. In contrast, Monte Carlo method only require experience in the form of sample sequences of states, actions and rewards and it is a way of solving the reinforcement learning problem based on averaging sample returns. An in depth review of the use of Monte Carlo methods for solving reinforcement-learning problems can be found in [94].

2.10 The use of Deep Convolutional Neural Networks in Reinforcement Learning

Convolutional neural networks have been a core component in modelling the manual process of laying up woven reinforcement as a reinforcement-learning task. This will become clearer in the next chapter. The main reason for this is the large number of state-action pairs that are present in building a model of the layup task. In this section of the chapter, a deep understanding of neural networks and convolutional neural networks, which is a variation of neural networks, will be developed. Developing this understanding is critical in understanding and putting together the reinforcement-learning model of the layup process.

2.10.1 Neural Networks

2.10.1.1 Introduction to Neural Networks

At a very fundamental level, a neural network is a simple and narrow implementation of the human brain's structure within a digital computer. While neural networks are the best architecture available as of today for mimicking the behaviour of the human brain, there is still a long way to go in mimicking the exact working mechanics of the human brain. The need for neural networks came about because of identifying that the human brain works in a very different manner to modern day digital computers. In a more mathematical sense, a neural network can be defined as a non-linear mapping from a fixed number of input parameters to a fixed number of output parameters that is learned through a pre-defined known true data set (inputs and outputs). In other words, it is a complex interpolation method inspired by the human brain's neuron structure. Neural networks are used as input-output mapping functions when it is not explicitly clear how to design a function to approximate the said mapping. Neural networks are also perceived as a universal function approximator. Once the learning process achieves an accuracy beyond a defined threshold, the trained network can be used to approximate outputs for input values outside of the training input values.

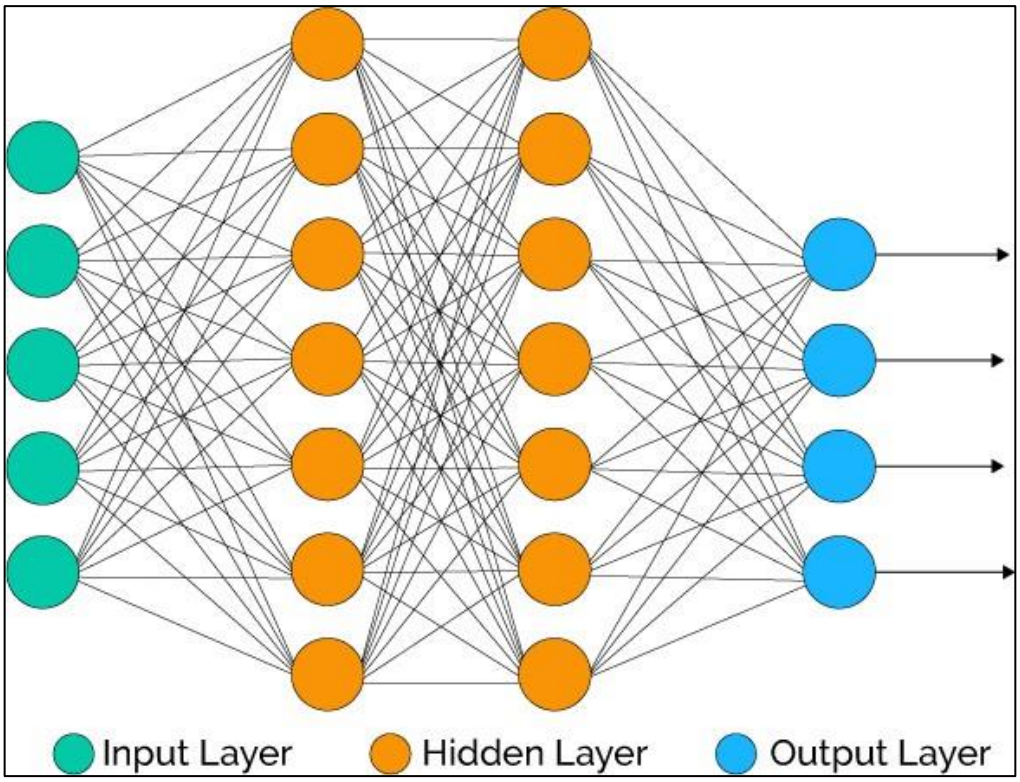


Figure 2.8 -Typical neural network architecture [96]

Figure 2.8 illustrates a typical neural network architecture. As evident from this figure, there are three types of layers in a typical neural network: (1) input layer, (2) hidden layers, and (3) output layer. Number of units (also called neurons) in the input and output layers are fixed prior to the design of the neural network architecture as they are determined by the training data set structure. Determining the number of hidden layers and number of units in each hidden layer is very much an art learned through trial and error. Output values from units of each layer are first multiplied by a weight matrix and a bias value is added to the result. Finally, the resulting values are passed through a non-linear function before they become the input values of the next layer. The values of units of hidden layers are initially set to random values. A technique named backpropagation is used to adjust these values until the accuracy of the neural network mapping of input values to output values of the training data set is beyond a threshold value. The trained network can then be used to produce approximated output values for new input values.

With this explanation of how neural networks operate, it can be said that neural networks resemble the human brain in two respects [97]:

1. Knowledge is acquired by the network by learning from its environment
2. Acquired knowledge is stored through interconnected synaptic weights

This is a very basic explanation on how neural networks operate. A deeper understanding of this process will be developed in subsequent sections of this chapter.

2.10.1.2 Digital Models of the Neuron

The fundamental building box of neural networks is the neuron. It is an information-processing unit. In Figure 2.9, the most commonly used nonlinear model of the neuron is illustrated. Three basic elements of this model are identified below:

1. A set of synapses characterised by their own unique weights. An input x_j, entering a synapse j, connected to a neuron k, is multiplied by the synaptic weight w_{kj}. The first subscript of the weight w refers to the neuron in question, and the second subscript to the input in question. Synaptic weights can take either positive or negative values.
2. A summation junction that adds all the weighted input values. This is a simple linear addition operation.
3. The activation function also referred to as the squashing function that limits the amplitude of the output from the summation unit. This is a nonlinear operation. Output from the activation function typically lies within [0, 1] or [-1, 1].

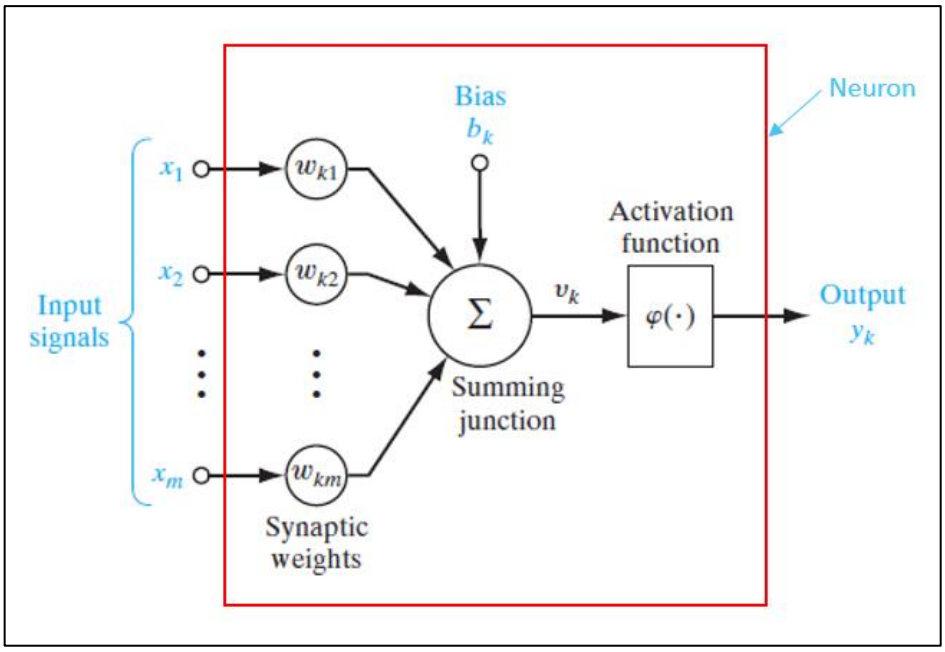


Figure 2.9 - Nonlinear model of the neuron [97]

In addition to the three main elements mentioned above, there is also an externally applied bias, denoted by b_k. The bias alters the net input of the activation function by a constant positive or negative value.

The input-output mapping of a neuron can be mathematically formulated as:

$$v_k = \left(\sum_{j=1}^{m} w_{kj}x_j\right) + b_k \qquad \textbf{Equation 2.10}$$

And,

$$Y_k = \varphi\,(v_k) \qquad \textbf{Equation 2.11}$$

Where φ (.) is the activation function and y_k is the output from the neuron k. This notation can be further simplified by defining $x_0 = +1$ and $w_{k0} = b_k$. The formula for v_k then becomes:

$$v_k = \sum_{j=0}^{m} w_{kj}x_j. \qquad \textbf{Equation 2.12}$$

Based on this new understanding, the model of the neuron can be illustrated as shown in Figure 2.10.

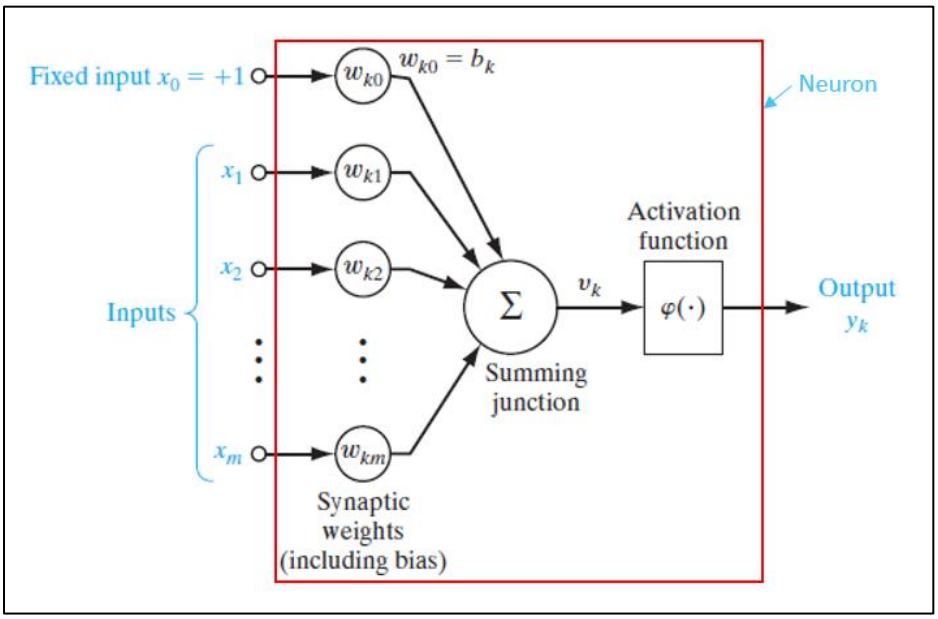


Figure 2.10 - Modified model of the neuron [97]

Types of Activation Functions

There are three basic types of activation functions:

1. Threshold Functions

For this type of function, as illustrated in Figure 2.11, the output from the neuron can take only two values: either 0 or +1. That is:

$$y_k = \begin{cases} 1, & v_k \geq 0 \\ 0, & v_k < 0 \end{cases} \qquad \textbf{Equation 2.13}$$

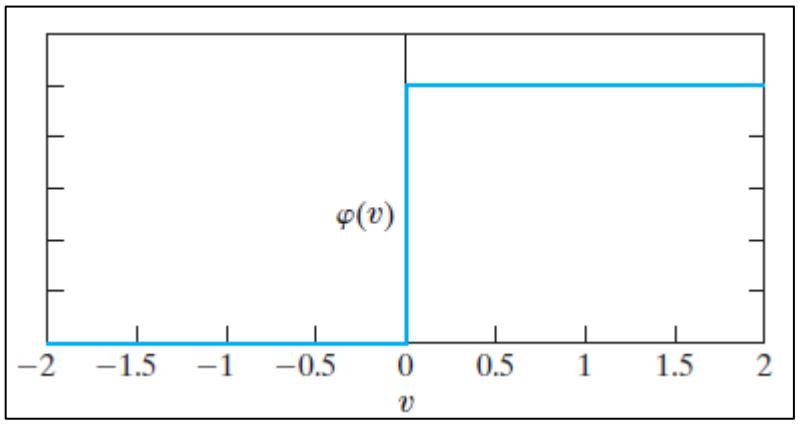


Figure 2.11 - The threshold activation function

2. Sigmoid Functions

The sigmoid function as illustrated in Figure 2.12, usually takes the form of an S shape. This function achieves a graceful balance between linear and non-linear behaviour. This has been the most commonly used type of activation function in constructing neural network architectures up until recently. Sigmoid functions works well when using only a handful of hidden layers. When used in deep neural networks (neural networks with a large number of hidden layers), sigmoid functions cause a mathematical problem called 'vanishing gradients' in backpropagation [98].

An example of a sigmoid function is the logistic function, defined mathematically as:

$$\varphi\,(\mathrm{v}) = \frac{1}{1 + \exp(-av)} \qquad \textbf{Equation 2.14}$$

Where a is the slope parameter of the sigmoid function. As illustrated in Figure 2.12, sigmoid functions with different slopes can be achieved by varying the slope parameter.

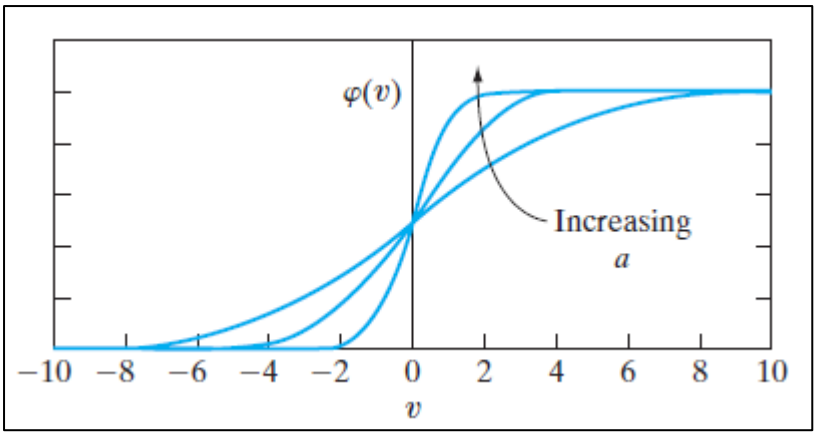


Figure 2.12 - The sigmoid activation function

Another commonly used sigmoid function is the tangent function, which is mathematically defined as:

$$\varphi\,(v) = \tanh(v) \qquad \textbf{Equation 2.15}$$

An activation function that can take both negative and positive values such as the tangent function has shown to yield more practical benefits compared to the logistic function that can only take positive values [97].

3. Rectified Linear Unit (ReLU) Functions

The ReLU computes the following function:

$$f(x) = \max(0, x) \qquad \textbf{Equation 2.16}$$

In other words, this activation function is a simple threshold at zero as illustrated in Figure 2.13. In [99], it has been shown that the use of ReLU units can accelerate the convergence of stochastic gradient descent by a factor of 6 compared to sigmoid/tanh functions. In addition, the ReLU units are computationally less expensive compared to

sigmoid/tanh functions [100]. However, pure ReLU units tend to die off during the training process as explained in [100].

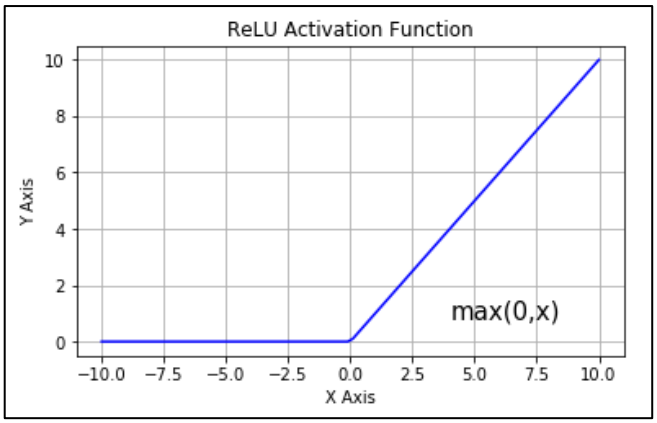


Figure 2.13 - The ReLU activation function [101]

2.10.1.3 Neural Network Architectures

There are three main fundamentally different neural network architectures: (1) Single Layer Feedforward Networks: In a layered neural network architecture, neurons are arranged in layers. In the simplest case, there is one input layer, one hidden layer, and one output layer. This is called a single layer neural network, since only the hidden layer performs computations. The network architecture is called feedforward because the computation or mapping from the input to the output only happens in one direction. Figure 2.14 illustrates this architecture.

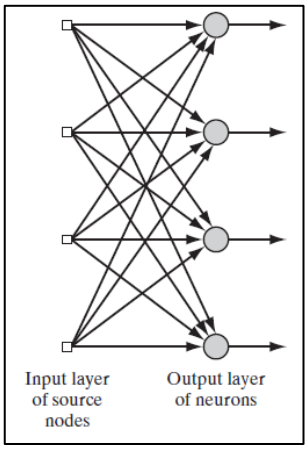


Figure 2.14 - Single layer feedforward neural network architecture [97]

(2) Multilayer Feedforward Networks: These network architectures contain two or more hidden layers. The function of the hidden layers is to estimate a function that maps the inputs to the outputs. By adding more hidden layers to the architecture, it achieves the ability to form functions that are more complex and are of higher orders. Figure 2.15 illustrates this architecture. The inputs are processed through the first hidden layer in a similar manner to a single layer feedforward network and the output from the first hidden layer is processed through the second hidden layer to finally produce the output.

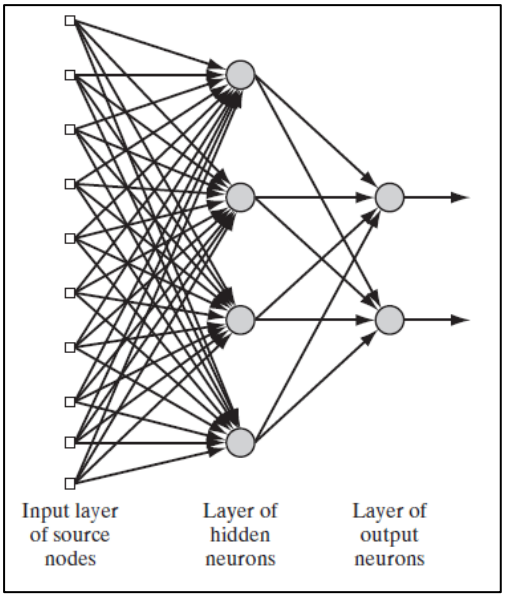


Figure 2.15 - Multilayer feedforward neural network architecture

(3) Recurrent Neural Networks: These architectures defer from feedforward networks in that they have at least one feedback loop. Since recurrent neural networks are not used in the work presented in this thesis, they will not be further reviewed. A detailed review of recurrent networks can be found in [97].

2.10.1.4 Learning Process of Neural Networks

Similar to how there are various methods that humans use to learn to achieve goals in the real world, there are various ways in which a neural network can learn to create an accurate input-output mapping. These learning methods can be broadly divided up in to two categories: (1) Learning with a teacher and (2) Learning without a teacher.

Learning with a Teacher

The technical term for learning with a teacher is 'supervised learning'. Figure 2.16 represents this learning process as a block diagram. In this context, the teacher has accurate knowledge of the environment in the form of input-output examples and the neural network has no knowledge of the environment.

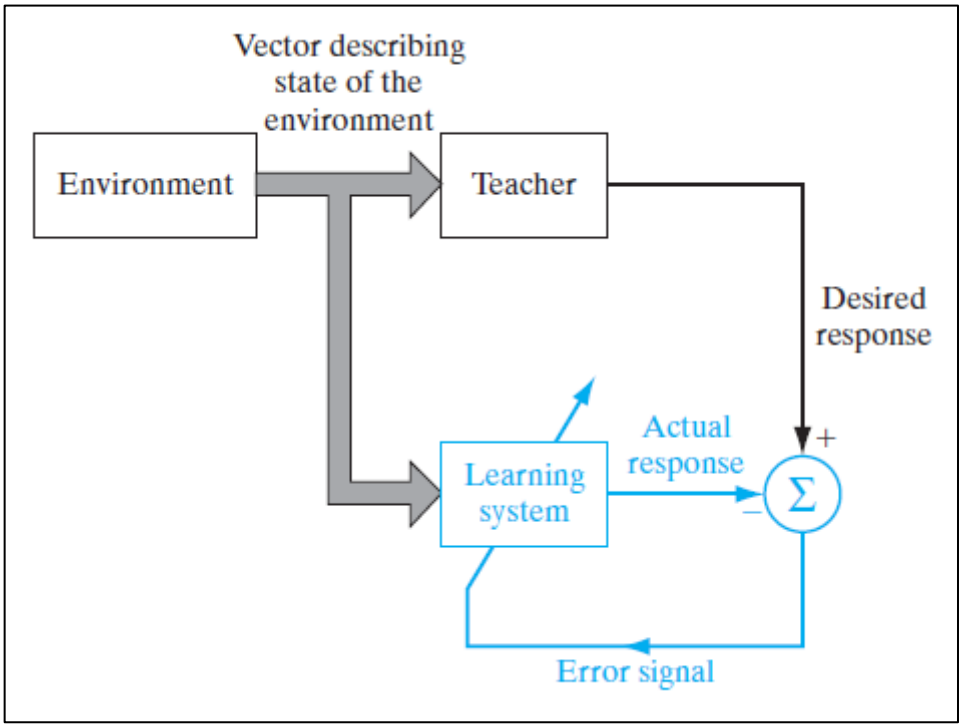


Figure 2.16 - Block diagram representing supervised learning

At the beginning of the learning process, weights of all synapses are set randomly. These weights are then adjusted using an error signal, which is a combination of the desired and actual responses of the neural network. This adjustment is carried out in a step-by-step manner until the neural network closely emulates the teacher's knowledge, or in other words, until the desired response and actual response becomes similar to each other for all training examples. When this condition is satisfied, the neural network can perform in the environment without the guidance of the teacher and should have the capability of providing accurate outputs to inputs that are outside of the training set of inputs.

The performance of the neural network is usually measured by using the mean squared error. The mean squared error is expressed as a function of all synaptic weights of the network and gradients of the mean squared error with respect to each weight is calculated at each error correction step. The mean squared error of the system can be visualized as a multidimensional error surface that is a function of every synaptic weight. The true error is averaged over all possible input-output examples or a representative batch of all examples in the case where there is a large number of examples. In minimizing this error, the objective is to find a set of synaptic weights that corresponds to the global minimum (or a good enough local minimum) of the error surface. This is achieved by backpropagation of error gradients from the output layer to the input layer using the chain rule and adjusting the synaptic weights in the process through gradient descent.

Learning without a Teacher

In this case, the network must learn the input-output mapping function without a teacher or in other words without labelled examples. There are two sub categories of this form of learning:

(1) Reinforcement Learning

A detailed description of reinforcement has already been provided in this chapter. Therefore, no further explanation is provided here.

(2) Unsupervised Learning

In unsupervised learning, the learning agent does not have access to labelled data or a reward system. The agent on its own must learn a model of the environment using only input data [102]. Unsupervised learning is considered to be very important by current experts in machine learning in reaching the ultimate goal of Artificial Intelligence, that is creating an Artificial General Intelligence, or in other words a human level intelligence [103] [104]. Unsupervised leaning is not further reviewed here, since it is not part of the work presented in this thesis. Detailed reviews on unsupervised learning can be found in [97], [105] and [106].

2.10.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks are very similar to neural networks in that they are also made up of learnable weights and biases [107]. They are used when the input is in the form of images and not simple numerical values. They still have the same mean squared error function and the same method of backpropagation of error derivatives are used to learn ideal weights in a step-by-step manner.

It can be argued that even an image input can be represented as a simple numerical vector input by using its RGB values of pixels, thereby using the same neural network architecture as described previously. This type of processing is also possible. However, by retaining the original structure of an image as the input and using a convolutional operation, which will be explained later, the performance of the neural network tends to be far superior and this allows a vast reduction of the total number of weights of the system, which reduces the computational cost significantly.

2.10.2.1 Architecture of CNNs

Regular neural network architectures do not scale ideally to images. Consider and image with medium size of 200x200x3. If a normal neural network was used to process this image, the input layer would contain 120,000 units and the number of weights between the input layer and the first hidden layer will be in the order of 120,000 x n, where n is the number of neurons in the first hidden layer. Therefore, we can imagine the total number of weights in the network adding up to an unmanageable amount very quickly. Moreover, having such a large number of weights could also lead to a phenomenon called overfitting of the neural network, which will stop the network outputs from generalizing to inputs outside of the training set.

CNNs take advantage of the fact that images have a constrained structure. Layers that are not fully connected in CNNs have neurons that are arranged in 3 dimensions: width, height and depth (the input layer of a CNN, which is not fully connected and represents an image with dimensions of 200x200x3 has dimensions: width – 200, height – 200 and depth – 3). In such layers, the neurons are only connected to a small region of the

previous layer. The final layer of a CNN is always a fully connected layer with height and width both being 1. Depth of the final layer depends on how many outputs we want to map the input to.

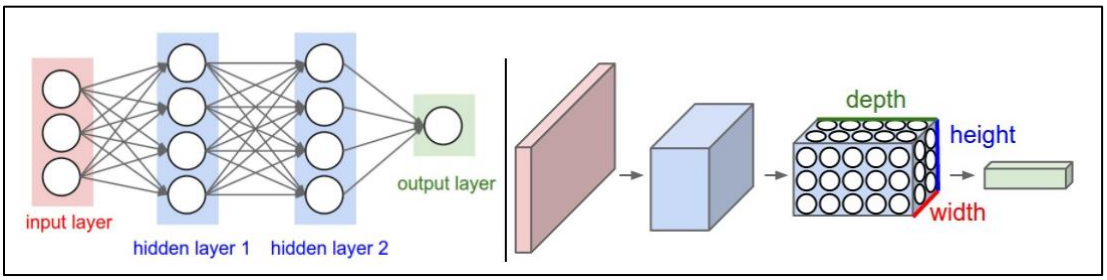


Figure 2.17 - Left: a regular 3-layer neural network. Right: a convolutional neural network, that arranges its neurons in three dimensions (width, height, and depth), as visualized in one of the layers [107] .

Every layer of a CNN transforms the 3D input volume to a 3D output volume of neuron activations. In the example shown in Figure 2.17, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, and Blue channels).

2.10.2.2 Layers Used to Build CNNs

CNNs are built by stacking three types of layers: (1) Convolutional layers, (2) Pooling layers, and (3) Fully-Connected layers. These layers are reviewed in detail below:

(1) Convolutional Layers

Overview: A convolutional layer is made up of a set of learnable parameters in the form of filters. Each filter has small spatial dimensions (width and height), but has a depth equal to the depth of the input volume. A typical first layer of a CNN might have an 'n' number of filters, with each of them having the dimension 5x5x3 (i.e. width and height are 5 and a depth of 3 because the input colour image usually has a depth of 3 due to its 3 colour channels: red, green and blue). In performing a forward pass through a convolutional layer, each filter is slid (or convolved) across the width and height of the input volume and the dot product between the entries of the filter and the input at every position forms the output volume. Each convolutional layer can have multiple filters and

each of them produces a separate 2 dimensional activation map. These activation maps are stacked along the depth dimension to form the output volume from that convolutional layer.

Local Connectivity: When the input in consideration is a high-dimensional image, each neuron is only connected to a local region of the input volume. The spatial extent of this connectivity is a hyper-parameter called the receptive field of the neuron (or the size of the filter). Along the depth axis, the extent of this connectivity is always equal to the depth of the input volume. In other words, connections are local in width and height, but full in the depth direction.

Spatial Arrangement: In this section, the volume of the output and its arrangement will be discussed. The size of the output volume is determined by three hyper-parameters:

(a) Depth: Depth of the output volume of a convolutional layer is equal to the number of filters used in that layer.

(b) Stride: Stride is the number by which the filter is slid. If it is decided to have a stride of one, then filters are moved one pixel at a time both in the height and width directions. Larger the stride smaller the output volume spatially.

(c) Zero-Padding: It is sometimes useful to pad the border of the input volume with zeros in order to control the size of the output volume spatially.

The spatial size of the output volume (O) can be expressed as a function of the input volume size (W), filter size (F), amount of zero padding (P), and the stride (S):

$$O = (W - F + 2P)/S + 1 \qquad \textbf{Equation 2.17}$$

Figure 2.18 illustrates an example of a convolutional operation. In this example, the input volume has original dimensions of 5x5x3. A unit zero padding has been applied, which makes the total volume of the input 7x7x3. The first convolutional layer has 2

filters, each with dimensions of 3x3x3. A stride of 2 is used. Using Equation 2.17, the spatial size of the output volume (O) must be:

$$O = (7 - 3 + 2*1)/2 + 1 = 3 \qquad \textbf{Equation 2.18}$$

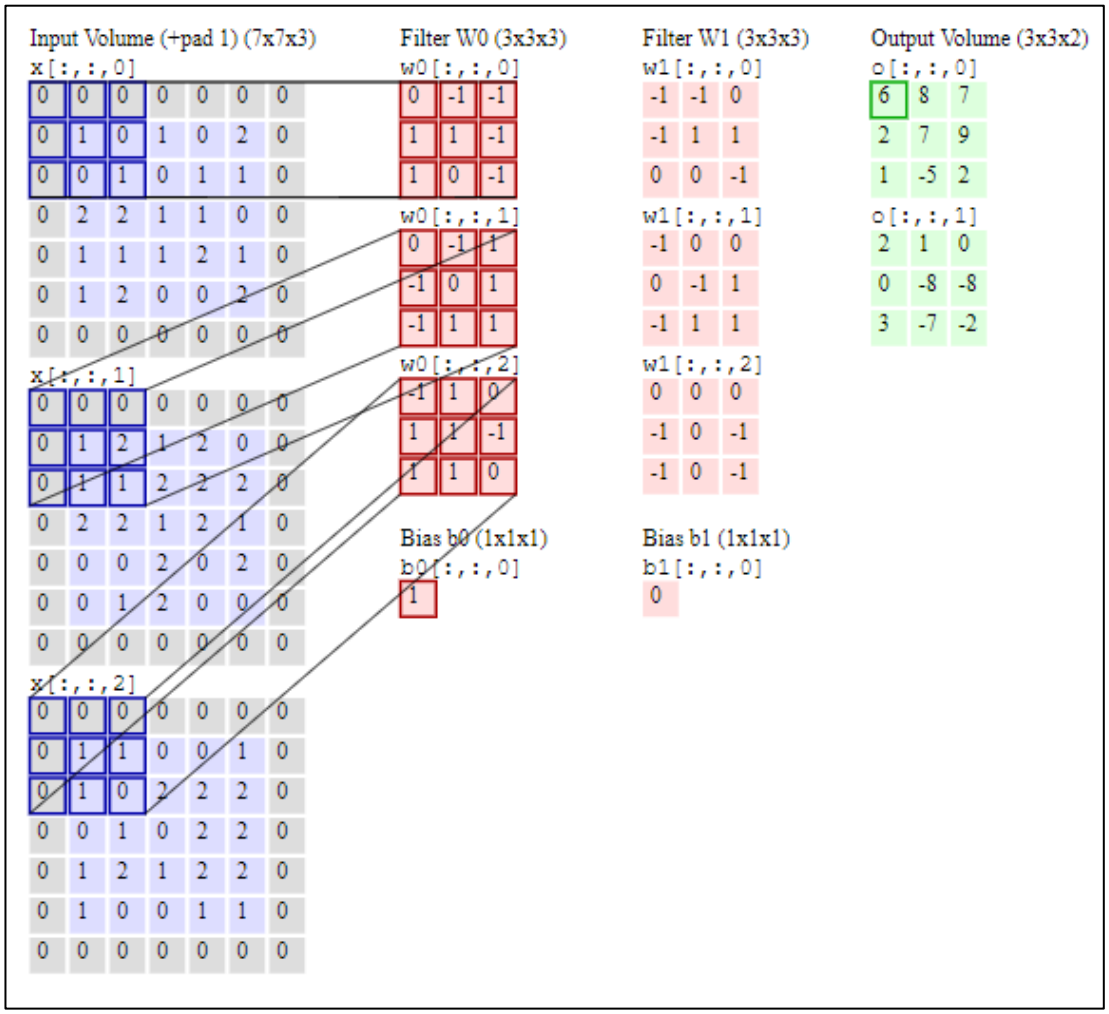


Figure 2.18 - An example of a convolutional operation [107]

(2) Pooling Layers: It is common to include a pooling layer between two consecutive convolutional layers in a CNN. Its function is to progressively reduce the spatial size of the output volume of a convolutional layer and thereby reduce the number of total weights and computational expense. It also helps with the problem of over fitting as discussed previously. Pooling layers do not affect the depth of the output volume from a convolutional layer. Pooling is essentially a down sampling operation and certain

amount of information is lost because of pooling. Due to this very reason, pooling is disliked by many researchers and recently there have been attempts made to discard their use [108]. Since, pooling layers are not used in the research work done in this thesis; they are not discussed further more here. A detailed description on pooling layers can be found in [107].

(3) Fully-Connected Layers: Fully connected layers have been discussed in detail in section 2.10.1 and hence will not be discussed here again.

2.10.3 Deep Q Reinforcement Learning Agent using CNNs

In this chapter, the concepts of reinforcement learning and convolutional neural networks are combined to present a single agent capable of learning how to layup multiple mould shapes. This is a general purpose learning algorithm, which belongs to the type of learning algorithms that forms the core of Artificial General Intelligence [109] that has eluded previous efforts ([95], [110] and [111]).

Recent advancements in deep neural networks ([112], [113] and [114]) have allowed neural networks to learn complex concepts using just raw sensory data. For the agent in question, a particular successful architecture named deep convolutional neural network [115] is used, which as explained previously, uses hierarchical layers of tiled convolutional filters to mimic the effects of receptive fields [116], inspired by Hubel and Wiesel's work on the visual cortex [117]. This allows building robustness to image based reinforcement learning systems in terms of natural transformations such as changes of viewpoint or scale.

The agent interacts with the environment through a sequence of observations, actions and rewards. The aim of the agent is to choose actions upon observation of the current state (presented to the agent as an image or images) in such a way to maximize the cumulative future reward. In designing this agent, the optimal action-value function is approximated using a deep convolutional neural network:

$$Q^*(s, a) = \max_\pi \mathbf{E}\,[r_t + \Upsilon r_{t+1} + \Upsilon^2 r_{t+2} + \ldots \mid s_t = s, a_t = a, \pi] \qquad \textbf{Equation 2.19}$$

Which is the maximum sum of discounted rewards achievable by a behavioural policy π, after making an action 'a' when presented with a state 's'.

The process of using deep convolutional neural networks in a reinforcement-learning environment can be summarised as:

1. The value function is parameterized as $Q(s, a, \theta_i)$ and is approximated using a deep convolutional neural network. Here, θ_i are the synaptic weights of the CNN at iteration 'i'. Note that θ_0, the initial synaptic weights are set randomly.

2. The agent then interacts with the environment at fixed time steps and collects a finite data set consisting of current states (s), actions (a), rewards (r) and subsequent states (s').

3. Q learning updates are then applied based on this finite data set. The loss function used by the Q-learning update at iteration 'i' is:

$$L_i(\theta_i) = (r + \Upsilon \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a, \theta_i))^2 \qquad \textbf{Equation 2.20}$$

Where θ_i^- are the network parameters used to compute the target at iteration 'i'. The target network parameters are only updated with the Q network parameters every C steps and are held constant between individual updates.

A more detailed explanation of this process is provided in the next chapter of this thesis, where a case study on the use of Deep Q Reinforcement Learning agents to generate optimum layup paths for a composite part is presented.

2.11 Summary

An immediate need for reinforcing the composite workforce, especially hand laminators, in the near future has been identified. Existing methods of training delivery

and knowledge transfer on hand lamination is yet to be standardized. Virtual Reality technology has been identified in this chapter as a platform of choice to address this issue.

Alongside this, it was also identified in this chapter, that in order to be able to automate the development of VR training simulations and to be able to provide standardized manufacturing instruction sheets to hand laminators, it is critical to be able to predict the ideal sequence of physical manipulations required to layup a given mould shape. Attempts made at using currently available drape simulators to predict the said ideal sequence of physical manipulations were reviewed and it was concluded that such previous attempts have failed to provide sufficient information required for shop-floor operatives to carry out layup operations. In searching for a solution to this limitation of drape simulators, their fundamental approach was re-evaluated, and it was concluded that perhaps the focus of drape simulators should shift more towards modelling the human learning process, coupled with a progressive deformation model for the reinforcement. The concept of reinforcement learning was introduced as a method to model the human learning process during hand layup. Reinforcement learning was defined as 'learning what to do – how to map states to actions – so as to maximize a numerical reward'. The concept of convolutional neural networks was then introduced and reviewed as a method of mapping states to actions in a reinforcement learning system. In conclusion, this chapter has provided the background knowledge and the theoretical foundation necessary to understand the case studies and that are presented in the next few chapters.

CHAPTER 3: Virtual Reality Training for Hand Layup: A Feasibility Study

3.1 Introduction

Research work presented in this chapter is a feasibility study on the use of VR technology and gamification on composites skills/knowledge transfer and to the knowledge of the author, this is the first time such an attempt has been made. The purpose of this feasibility study is to gain an understanding on the suitability of VR technology in composite layup training. Process chosen for the feasibility study is the layup of a flat carbon fibre composite panel using unidirectional prepreg material in a typical composite clean room environment. Work presented in this chapter has been adapted from previous work carried out by the author in [56].

3.2 Actions Involved in the Layup of a Flat Prepreg Panel

The layup of a prepreg flat panel is a simple process compared to other composite manufacturing processes. The actions involved in this process are straightforward and cannot be categorized as fine craft techniques. The set of actions used and the order in which they are performed may vary between different composite manufacturing centres to best suite their facilities. The best practice guidelines used at the University of Bristol's composite clean room in performing a prepreg flat panel layup were identified following a consultation with its head composite technician. This set of actions is tabulated in Table 3.1 [118]. Graphical representations of these actions as designed in to the VR simulator is illustrated in Figure 3.1. Graphical representations of these actions being performed by the author is illustrated in Figure 3.2. As it is difficult to use images to understand the design of the VR simulation and the real actions them self, a video of the VR simulation and a video of the actions being performed by the author are provided in [119] and [120]. This set of actions is based on a layup sequence of ([0, 90, 45, -45]

s). Layup of only the first four prepreg layers are considered, since the layup procedure for the next four layers is identical to that of the first four.

Table 3.1 - List of actions in the layup of a prepreg flat panel.

Action	Description
1	Pick up and wear a lab coat.
2	Pick up and wear latex gloves.
3	Pick up 2 metal L shapes, a Stanley knife and a roller and place them on the layup table.
4	Pick up the stack of prepreg from the defrosting station and place it on the table.
5	Align the first layer using the 2 L shapes.
6	Remove top backing paper of the first layer in the fibre direction (Use the Stanley Knife).
7	Tick off the first layer.
8	Remove the bottom backing paper of the 2nd layer in the fibre direction.
9	Place the provided release film on the first layer leaving a small gap on the top.
10	Lay the 2nd layer on top of the 1st one with alignment checks and gradual release film pull out.
11	Consolidate the first two layers using the roller.
12	Remove the top backing of 2nd layer in the fibre direction.
13	Tick off the second layer.
14	Repeat the layup procedure for the 3rd layer.
15	Repeat the layup procedure for the 4th layer.

Action	Description
16	Move the laid up four layers to the consolidation station.
17	Place the provided release film on top of the four layers.
18	Pick up a preformed frame and place it on top of the four layers.
19	Connect a vacuum pump to the frame to debulk the four layers.

(Table 3.1 continued)

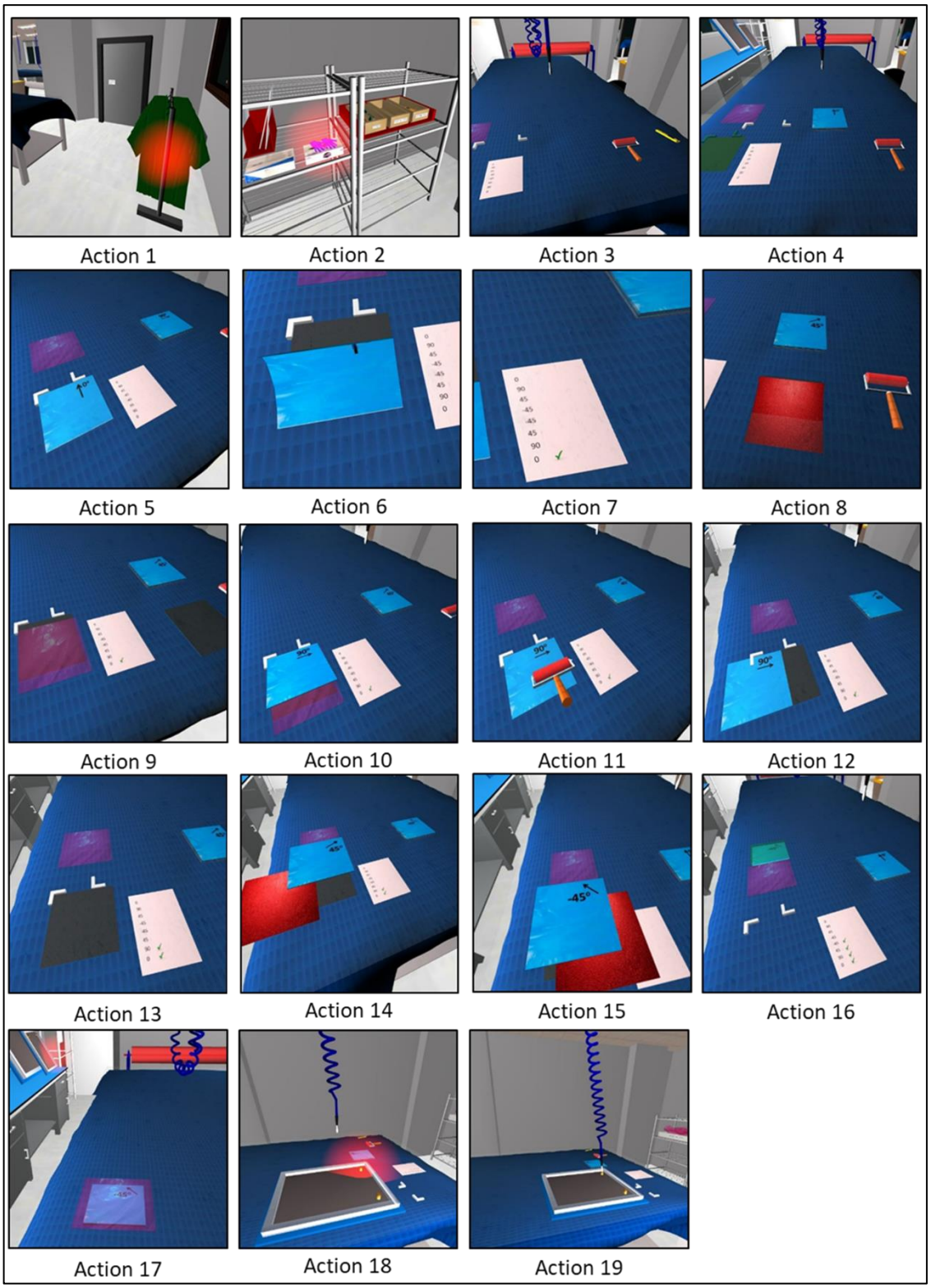


Figure 3.1 – All actions of the virtual reality simulation used to train the layup of a flat prepreg panel. (Refer to Table 3.1 for action descriptions)

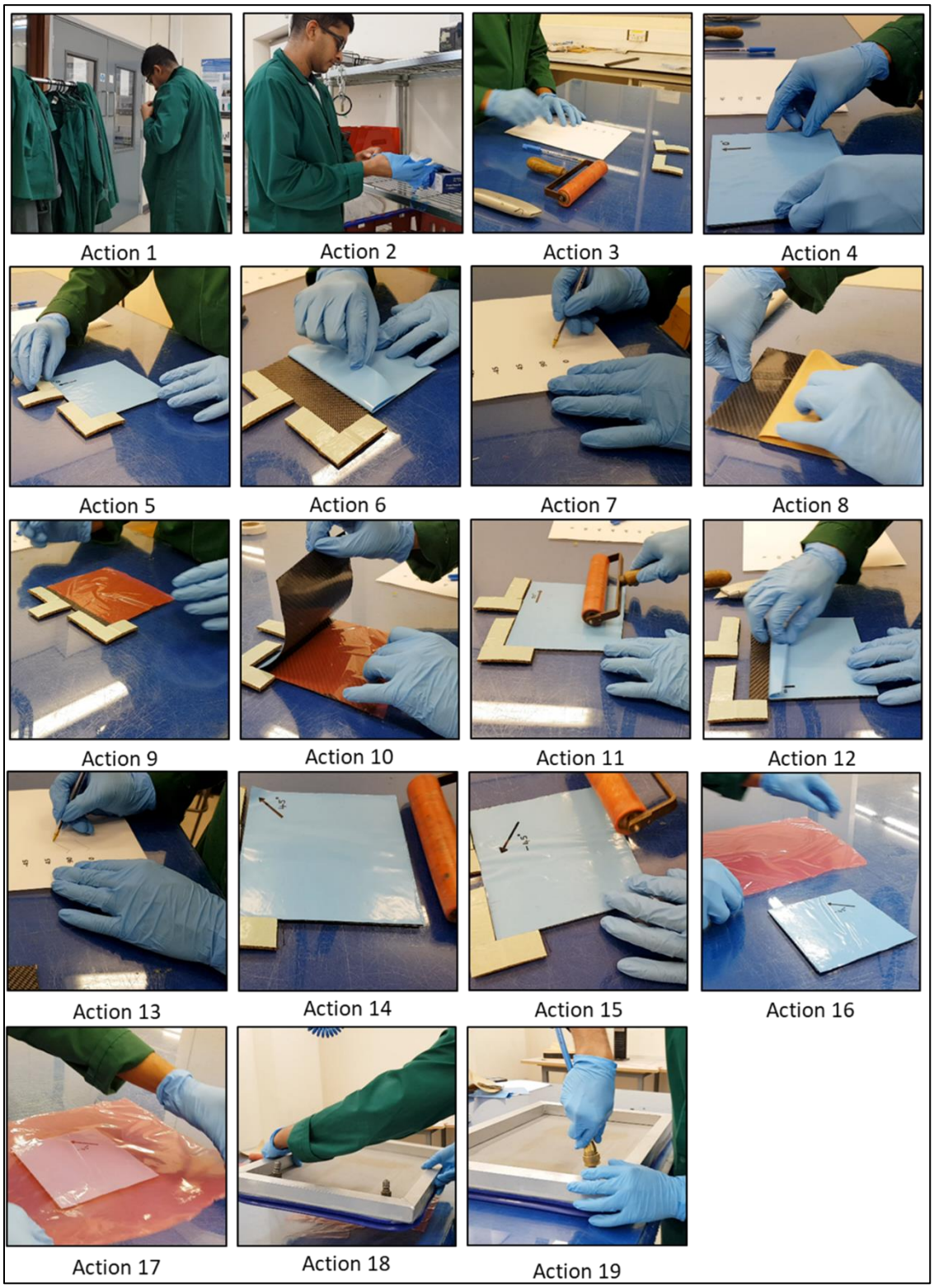


Figure 3.2 - Illustration of all actions being performed by the author. (Refer to Table 3.1 for action descriptions)

3.3 Current Method of Delivery of Training

The head composite technician provides a presentation to groups of novice laminators, in which each action outlined in Table 3.1 is explained in detail. The presentation is followed by a demonstration of the layup procedure. There are many occasions where there exists a significant time gap between this initial induction and the first time novice laminators perform a layup. This leads to many errors in the layup procedure. Even experienced laminators regularly omit certain simple actions such as wearing a lab coat or gloves prior to performing a layup. Use of a VR training guide prior to performing a layup could lead to significant reductions in these simple mistakes via memory reinforcement and repetition. It will also reduce the workload on the technician by minimizing the need for retraining.

3.4 Software and Hardware Requirements

Main software requirements for the design of the VR system are 3D graphics design and game mechanics design. All 3D graphics required for the virtual environment were designed using Autodesk 3ds Max (chosen due to the author's previous experience with it and its compatibility with the chosen game engine. Other choices include Cinema 4D, Modo, Maya and Blender). Game mechanics were designed using the 'Unity 3D' game engine. The 'Unity 3D' game engine has a relatively short learning curve compared to others. In addition, it is a multi-platform game engine, which allows the designed simulator to be used in iOS, Android, or Windows mobile platforms with minimal effort.

Four main hardware components were used in the design of the VR system: high performance android smartphone (HTC One (M8)), DUROVIS DIVE hands-free VR headset (the only commercially available hands-free smartphone VR headset at the point of system design), Bluetooth game controller, and stereo headphones. Functions of the main hardware components used are outlined in Table 3.2.

Figure 3.3 illustrates the complete system in use and Figure 3.4 illustrates the interface between software and hardware used in the system.

Table 3.2 - Hardware in the VR system for training of layup of a flat prepreg panel and their functions.

Hardware Component	Function(s)
Smartphone	Computer processing (Graphics, game mechanics, input from Bluetooth controller and audio output to headphones), rendering of individual images to left and right eye to create the stereoscopic 3D effect and user head orientation tracking (using inbuilt gyroscope and accelerometer).
VR Headset	Converting the smartphone into a head mounted display and concentration of field of view of each eye on to the required image area of the smartphone display using biconvex lenses.
Bluetooth Controller	Primary input device to the VR system.
Headphones	Provide audio instructions in guiding the user through the layup procedure.

Figure 3.3 - Delivery of the training aid to the user illustrated by the author.

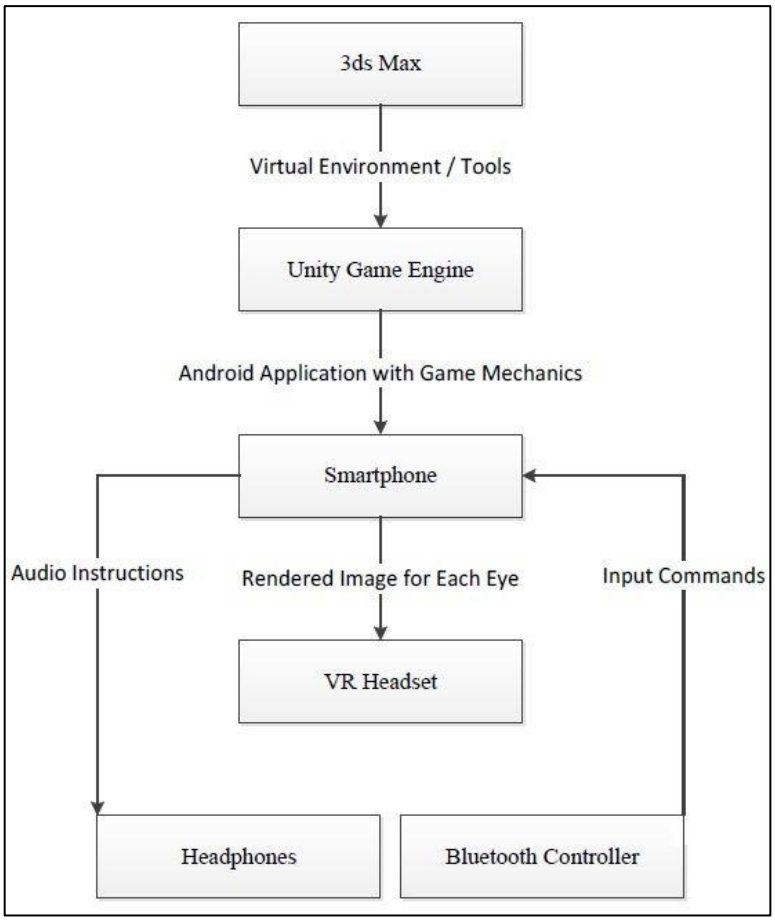


Figure 3.4 - Software/hardware interface of the virtual reality system.

3.5 Design of the Virtual Environment

The virtual environment was designed to represent the composite clean room at the University of Bristol. The floor plan of the composite clean room at the University of Bristol was measured and multiple photographs of the clean room were also taken by the author to capture as much information as necessary to create a 3D model of the clean room. This was achieved in less than 2 hours. The graphics design stage was completed within 3 weeks, with working hours of 9 am to 5 pm. As previously mentioned, all graphics were designed in 3ds Max and then exported in to the Unity game engine. Optimization of graphic performance is critical, especially when the design platform is a mobile device, where capacities of the Central Processing Unit (CPU) and Graphics Processing Unit (GPU) are relatively low compared to desktop processors. Graphics optimization is also critical to ensure that the smartphone screen refresh rate

is maintained at maximum. The maximum refresh rate of the smartphone used is 60Hz. This is not ideal for VR applications, as it is only at 90Hz that 95 to 99 percent of people do not notice the refresh rate anymore [121]. To address this issue, best practice guidelines on graphics optimization for mobile devices have been adopted in the design of the virtual environment [122].

Figure 3.5 provides a comparison between the actual clean room and its designed virtual model giving an idea of the level of detail captured.

Figure 3.5 - Comparison between the actual clean room (left) and the virtual clean room (right) to demonstrate the level of detail captured.

3.6 Design of Game Mechanics

The VR simulation has adopted the free-roam game genre and placed the user in the virtual environment with a first person view. All objects within the virtual environment are referred to as “Game Objects” in the Unity game engine. The user is represented by an ellipsoidal object with a capsule collider (green wire frame) attached to it, and will be referred to as ‘player’ from this point onwards. A capsule collider is used to detect the player approaching specific tools within the virtual environment (Refer to Figure 3.6, Figure 3.7 and Figure 3.8 for further clarification of this concept) .Two cameras placed at the same vertical level (Y), Refer to Figure 3.6, but 6cm apart (average distance between left and right eye of a human) horizontally (X); are made dependent game

objects of the player. It is these two cameras that are used to render separate images to each eye of the user to create the stereoscopic 3D effect. Movements of the player in the horizontal plane (XZ) of the virtual world are controlled by the Bluetooth controller. Since the two cameras are dependent on the player, camera position in the horizontal plane changes relative to the player. Orientations of the cameras are synced to orientation of the user's head using the gyroscope and accelerometer of the smartphone. The player is only given freedom of movement in the virtual environment. The player is forced to perform the set of actions outlined in Table 3.1. The player is provided spatial directions in the virtual world using a red hotspot (Refer to Action 1 in **Error! Reference source not found.**).

Scripting is at the core of the game mechanics design. Scripting language used was "C Sharp". Use of an efficient scripting architecture is paramount in reducing the workload on the CPU. The scripting architecture should be such that the number of calculations/checks done per frame by the CPU is at a minimum. A separate script was written for each action. Using a global static integer variable named "GameIndex" initiated in a separate script named "GameController", each action related script is activated/deactivated depending on the current action to be completed by the player. Another script named "AudioController" plays audio instructions based on the action to be performed and this is also linked to the "GameIndex". A simplified version of the scripting architecture is illustrated in Figure 3.7 using completion of Action 1 as an example. Refer to Figure 3.8 for further clarification.

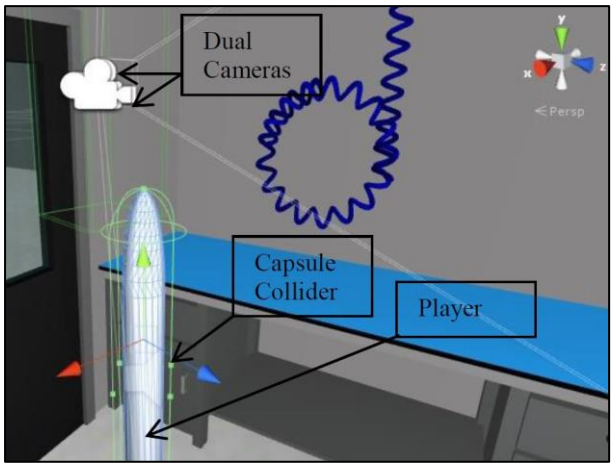


Figure 3.6 - Setup of the player in the unity gaming engine showing the components (capsule collider and stereoscopic cameras) attached to it.

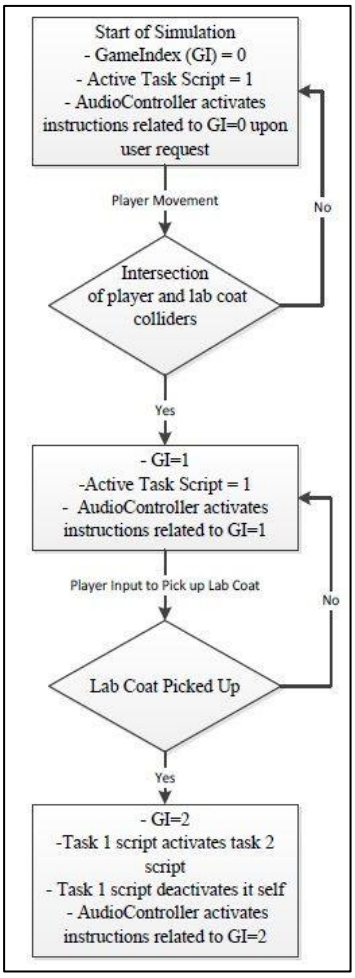


Figure 3.7 - Simplified scripting architecture expressed using progression from task 1 to 2 in the simulator.

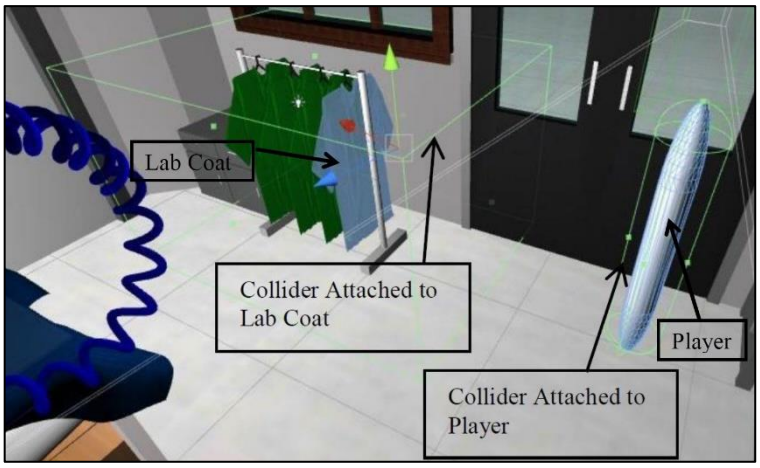


Figure 3.8 - Illustration of collider interaction between the player and the lab coat.

3.7 Evaluation of the Training Aid

Evaluation of the training aid was carried out in two stages. The first stage involved evaluating the training aid, using a pilot group in order to obtain feedback on technology acceptance and future improvements. The pilot group consisted of 30 research students who have previously carried out a layup of a composite flat panel in a clean room. Each student was asked to play through the simulation three times. Prior to the beginning of the simulation, a brief introduction to it and the Bluetooth controller layout were explained to each student. At the end of each testing session, students were asked to fill out a technology acceptance questionnaire to evaluate two main aspects of the training aid:

(i) Simulator sickness issues (adapted from [123])

(ii) Simulator usefulness (adapted from [76])

Acceptable levels of discomforts listed in Figure 3.9 depend on the time of use of the simulator, and on average 23 minutes were spent by each research student in completing the simulation three times. Two out of the thirty (6.7%) students opted out of the simulation due to nausea related issues. It is reasonable to set the maximum acceptable level of these discomforts to halfway between slight and moderate (a value of 2.5). Figure 3.9 confirms that all discomforts measured are well below this acceptable level. It should be noted that feedback on eyestrain will also depend on the individual

adjustment of the lenses of the VR headset. Simulator usefulness questionnaire yielded mixed results as illustrated in Figure 3.10. While the first five questions received highly positive feedback, majority of the pilot group's opinion on the use of such a simulator as a stand-alone training guide was negative. This was expected at the design stage of the simulator, as the use of a handheld game controller is unable to transfer physical skills to the user of the simulator. However, the author decided to go ahead with this experiment, since the objective here was to carry out a quick feasibility study on the suitability of VR technology to train laminators before spending a significant amount of financial resources on state-of-the-art VR equipment. Feedback from this stage of testing can also be used to optimize the layout and positioning of tools within the clean room. Time based analysis can be carried out for different layouts and tool positioning in order to find the optimum configuration.

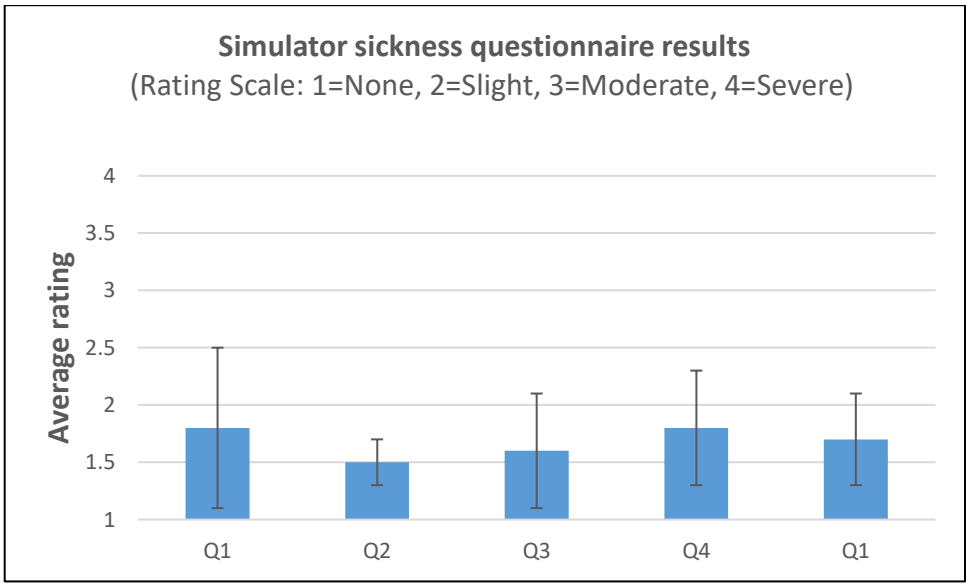
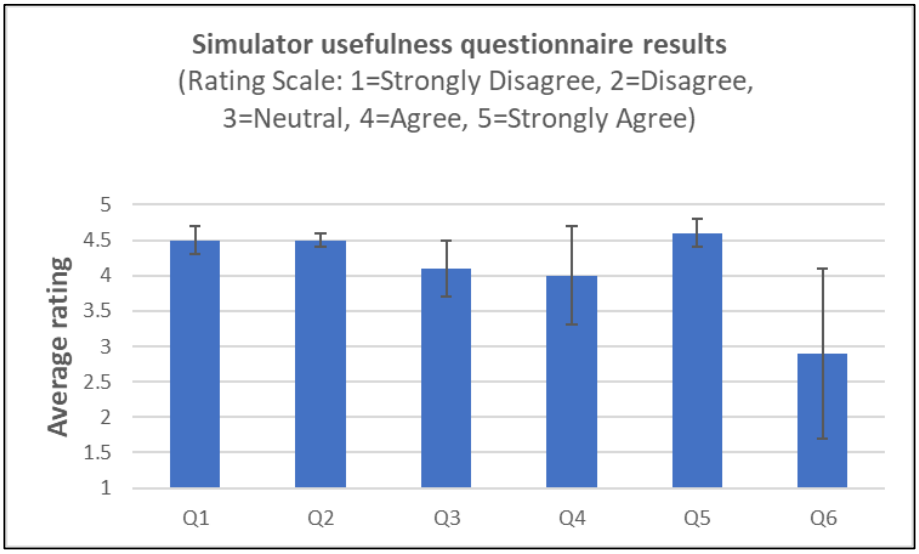


Figure 3.9 - Simulator sickness questionnaire results (Error bars represent standard deviation)



Q1 - This simulator helped you to learn the layout of the clean room.
Q2 - This simulator helped you to learn the whereabouts of tools in the clean room required for a composite panel layup.
Q3 - This simulator helped you to learn how to use the above mentioned tools in the layup procedure.
Q4 - You will be more confident in performing a composite panel layup after using this simulator, compared to only having access to written and video instructions.
Q5 - This simulator helped you to learn the steps in the layup of a flat panel.
Q6 - This simulator alone is sufficient to learn the layup procedure (i.e. you don't need an instructor helping you if you were performing the layup for the first time).

Figure 3.10 - Simulator usefulness questionnaire (Error bars represent standard deviation)

Objective of the second evaluation stage was to test the effectiveness of the simulator as a knowledge/skills transfer tool on novice laminators (i.e. candidates who had no prior knowledge on composites or composite layup). Two groups (A and B) of 10 candidates each were selected. Each candidate of group A was individually briefed on how to perform a composite layup, and was then made to watch a video of the layup being carried out in the clean room [120]. Immediately after the briefing, the candidate was asked to perform a composite flat panel layup in the clean room following the steps identified in Table 3.1, with it being set up identical to that in the simulation. Each action performed by the candidate was observed and given a score of 1 if it was accurate and in the correct order, as it was designed. A score of 0 was given to any action performed inaccurately or in the wrong order. Refer to Appendix: Table 2 for the scoring criteria used. Each candidate of group B was provided the same briefing and performed the

same layup but was asked to play through the VR simulator instead of being provided with video instructions.

It can also be argued that some of the actions listed in Table 3.1 are more critical to a successful layup compared to others. Putting aside the success of the final layup in terms of accuracy, certain actions that are related to health and safety could be even more significant for organizations. This type of ranking of actions in terms of importance could lead to a more complicated scoring system. However, since the purpose of this study is to study the effectiveness of VR technology as a skills transfer tool, the author has decided to give each action the same importance.

The total score for each candidate of both groups A and B were calculated and individual scores as a percentage of the maximum possible score were determined and are illustrated in Figure 3.11. Through an independent T-Test [124], it can be shown that Group B (Mean = 88.8, SD = 9.7), who received VR training, compared to Group A (Mean = 63.0, SD = 13.1), who received video instructions, experienced a significantly shorter learning curve, $t(18) = 5.0$, $p<0.001$.

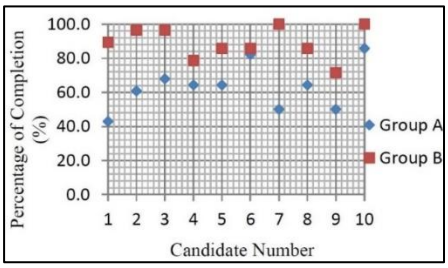


Figure 3.11 - Individual percentage scores from groups A (video training) and B (virtual reality training) on the skills/knowledge transfer test

3.8 Summary

The designed VR training aid, in its current form is an effective knowledge capture/transfer tool. However, so far has limited use in physical skill transfer. As a knowledge capture/transfer tool, it has shown potential in the reduction of the learning curve of novice laminators as an increase in task completion of greater than 20% was observed when test candidates used the designed VR training simulator as opposed to

when candidates were just provided with video instructions. The specific method of delivery of training tested in this chapter, in the form of head mounted VR gear was also well received by first time users, with only 6.7% of test candidates opting out during testing due to discomfort related issues and the feedback received was mainly positive.

Future developments to the tool presented in this chapter with hand movement integrations and haptic feedback via the use of data gloves and/or optical hand tracking devices will enable the transfer of physical skills and demonstration of layup of more complex parts. Some of these issues and suggestions are addressed in the next chapter of this thesis.

CHAPTER 4: VR Training for Hand Layup: Case Study with a 3D Geometry

4.1 Introduction

In this chapter, a case study on the use of VR technology with integrated hand tracking for hand lamination training is presented. As concluded in the previous chapter, hand tracking might be essential in trying to use VR as a platform for hand lamination training, when it comes to more complex shapes than flat plates that require more complex manipulations than simply picking up and placing flat sheets of prepreg on top of each other. In attempting to test this concept, a Leap Motion controller was combined with an Oculus Rift Development Kit 2 to create a VR training simulation for the layup of a 3D mould shape. Design criteria and findings from this case study are discussed in this chapter.

4.2 Design Features of the VR Training Simulation

4.2.1 Mould Shape and Features

The dimensions of the mould shape used for the VR training aid is illustrated in Figure 4.1. This particular mould shape was chosen since multiple expert laminators were made to layup this shape in [22] and the author had direct access to video recordings of this study. In addition, physical manipulations required to layup this mould shape were judged simple to be programmed in to a VR simulation with hand tracking.

Key features of this mould have been labelled in Figure 4.2, in order to make the explanation of the identified layup sequence for this mould in section 4.2.2 clearer. As evident from Figure 4.2, there are four edges and three grooves in the mould. The layup surface has been divided in to 9 sections labelled from A to I.

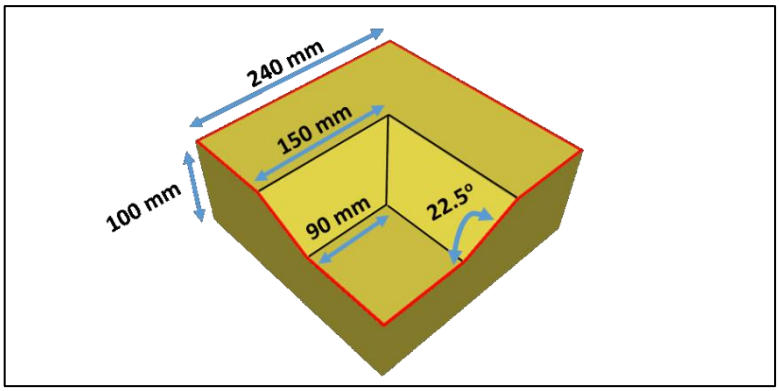


Figure 4.1 – Dimensions of the mould shape used for the VR training simulation

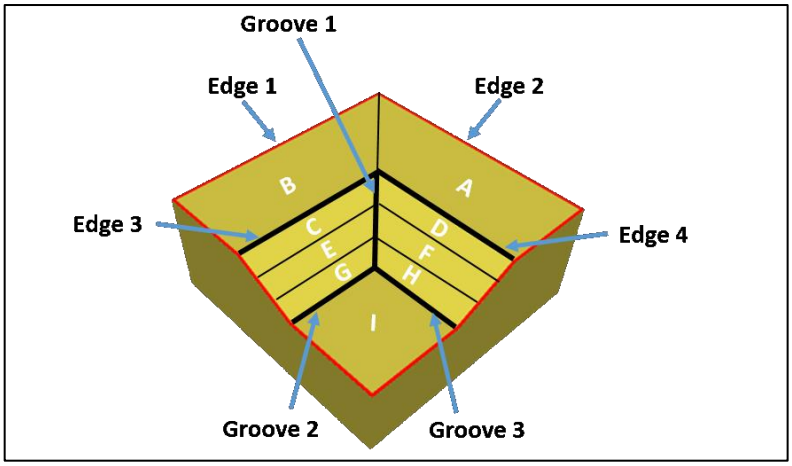


Figure 4.2 - Features of the mould used for the VR training simulation

4.2.2 Layup Sequence

The layup sequence for this particular mould was determined by analysing videos of two expert laminators laying it up, acquired from [22], which was then followed up by multiple attempts by the author trying to mimic the same layup sequence to confirm the validity of the identified layup steps. The "Dibber" [47] was also incorporated in to the layup sequence where appropriate to explore the use of layup tools in a VR training simulation. It is worth noting that the layup sequence used for this case study is just one of many possible layup sequences that can be used to layup this particular mould shape. The objective of this case study is not to figure out the best possible layup sequence for this particular mould shape. Instead, it is to explore how effective a VR training simulation is in teaching a predefined set of physical manipulations to a laminator.

The sequence of actions used in designing the VR training simulation is outlined in Table 4.1. Graphical representations of these actions is provided in Figure 4.3.

Table 4.1 - List of actions in the layup of the 3d mould shape illustrated in Figure 4.1

Action no.	Description (RT = Right Thumb, LT = Left Thumb, RMF = Right Middle Finger)
1	Place the prepreg on the mould with Edge 1 & 2 aligned with its two adjacent edges.
2	Press the prepreg on to the mould using RT and secure one of its edges to Edge 1 using LT.
3	Press the prepreg on to the mould using LT and secure one of its edges to Edge 2 using RT.
4	Press the prepreg on to the mould using LT and use the Dibber to secure it on to section A.
5	Press the prepreg on to the mould using LT and use the Dibber to secure it on to section B.
6	Press the prepreg on to the mould using RT and use LT to secure it on to Edge 3.
7	Press the prepreg on to the mould using LT and use RT to secure it on to Edge 4.
8	Press the prepreg on to the mould using LT and use RMF to secure it into the section of Groove 1.
9	Press the prepreg on to the mould using LT and index fingers and secure it on to section C using the Dibber.
10	Press the prepreg on to the mould using LT and index fingers and secure it on to section D using the Dibber.
11	Press the prepreg on to the mould using LT and use RMF to secure it into Groove 1.
12	Press the prepreg on to the mould using LT and index fingers and secure it on to section E using the Dibber.
13	Press the prepreg on to the mould using LT and index fingers and secure it on to section F using the Dibber.
14	Press the prepreg on to the mould using LT and use RMF to secure it into Groove 1.
15	Press the prepreg on to the mould using LT and index fingers and secure it on to section G using the Dibber.
16	Press the prepreg on to the mould using LT and index fingers and secure it on to section H using the Dibber.
17	Press the prepreg on to the mould using LT and use RMF to secure it into Groove 2.
18	Press the prepreg on to the mould using LT and use RMF to secure it into Groove 3.
19	Use the Dibber to secure the prepreg on to section I.

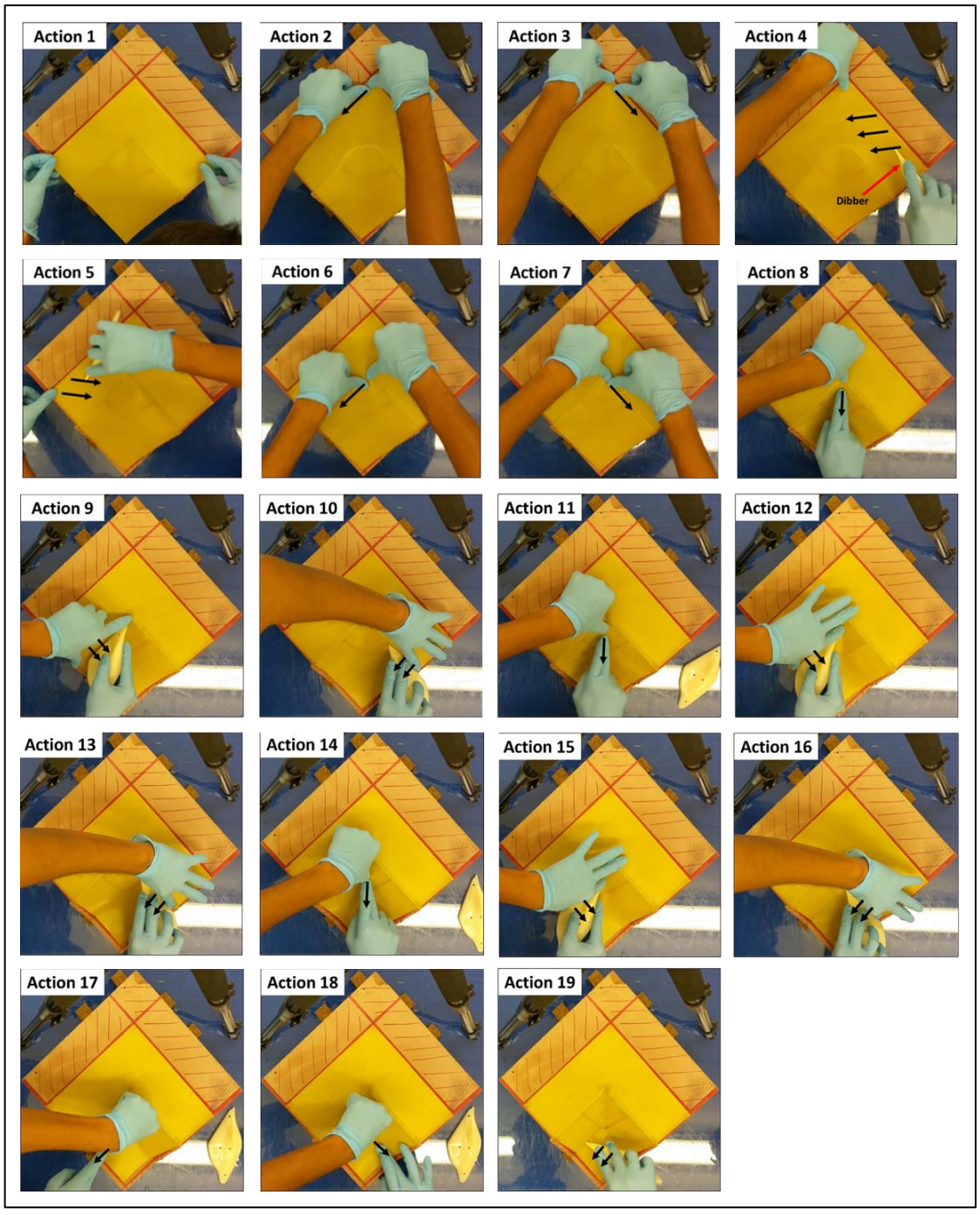


Figure 4.3 - Graphical representations of actions in laying up the 3D mould

4.2.3 Hardware and Software

Two main hardware components were used for the design of the VR training simulation:

1. Oculus Rift Development Kit 2: As illustrated in Figure 4.4, there are two main components included in the Oculus Rift Development Kit 2: a head mounted display (left) and an infrared sensor (right). The head mounted-display contains two separate screens for each eye. When strapped on, the result is a 3D effect that creates the illusion of a virtual environment [125]. The head mounted display (HMD) is covered with a group of infrared lights that are invisible to the naked eye. Light emitted from these infrared lights can be detected using the infrared sensor, which must be placed with unbroken line of sight between it and the HMD. This interaction between the infrared sensor and emitted light is used to track the rotation and movement of the HMD in 3D space. This information is then used to update the views of the displays accordingly. In addition to the infrared sensors, the HMD also includes an accelerometer, gyroscope and magnetometer that aids with head tracking.

This particular VR headset was chosen, since it was the only commercially available VR headset at the point of design of this VR straining simulation that was capable of coupling with hand tracking hardware.

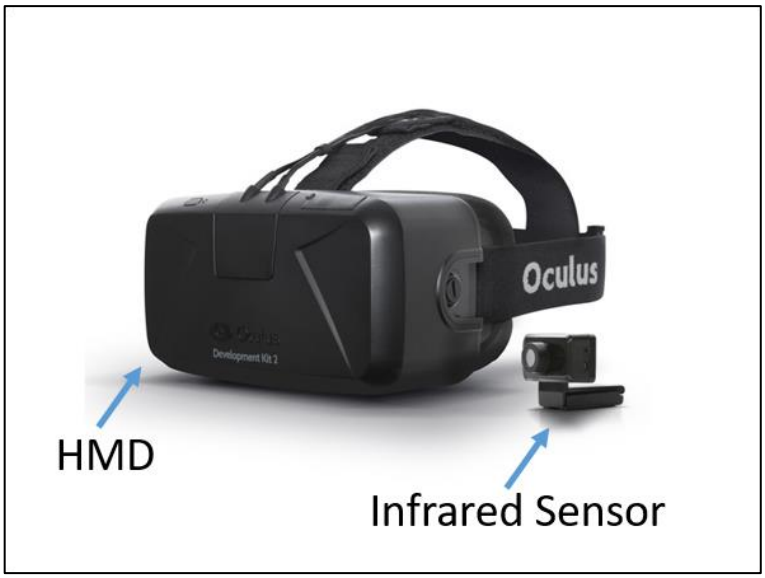


Figure 4.4 - Components of the Oculus Rift Development Kit 2 [126]

2. Leap Motion Hand Tracker: The Leap Motion is a small USB device consisting of mainly two cameras and three infrared LEDs. They are used to track infrared light with a wavelength of 850 nm [127]. This device has an interaction space of 8 cubic feet, which takes the shape of an inverted pyramid as illustrated in Figure 4.5. It is capable of tracking both hands and all 10 fingers as they move through the 3D space in front of the sensor. While the Leap Motion controller is very much a stand-alone product, with the introduction of VR headsets such as the Oculus Rift, the Leap Motion controller has opened up new possibilities in the realm of Virtual Reality as a hand tracker. As illustrated in Figure 4.6, the Leap Motion controller can be attached to the front of the Oculus Rift to be used as a hand tracker.

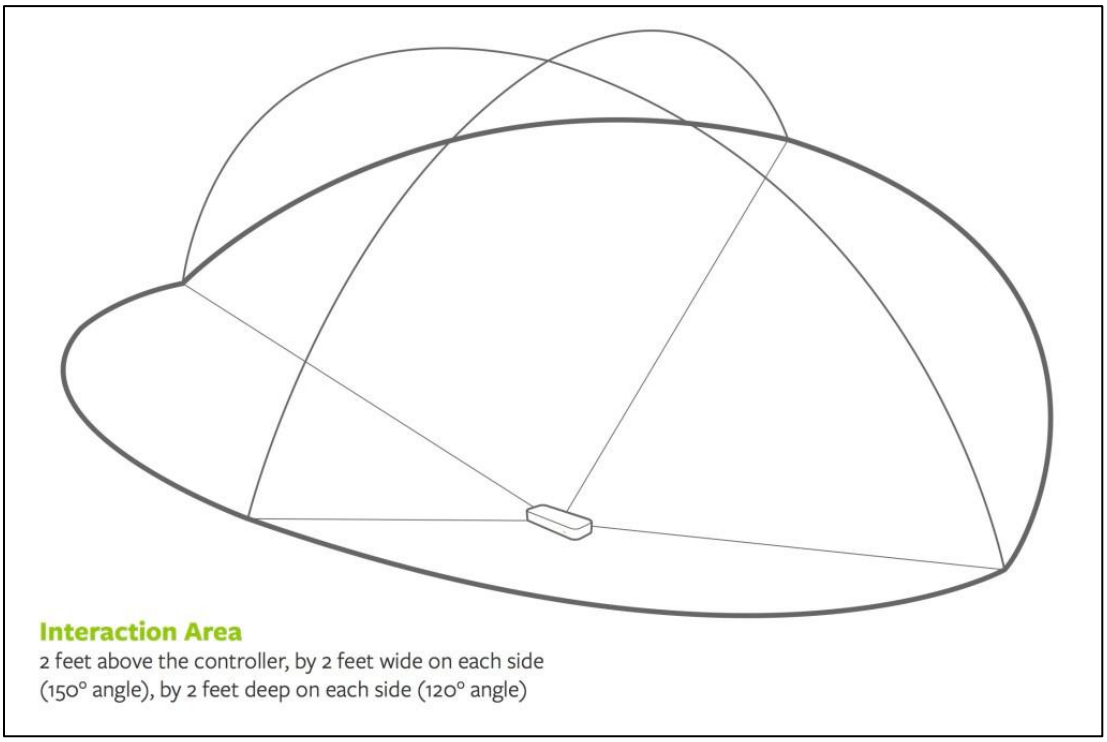


Figure 4.5 - Interaction area of the Leap Motion controller [127]

Figure 4.6 - Leap Motion controller attached to the Oculus Rift

The entire VR training simulation was designed and run through the Unity 3D gaming engine, since it provides straightforward cross platform support between both the Oculus Rift and the Leap Motion controller.

4.3 The VR Training Simulation

4.3.1 Simulation Environment

The training simulation is designed to take place inside a closed room environment, with the mould placed on a table placed at the centre of the room. A layup instruction player is placed on the wall in front of the user to aid in the training process. A simple model of hands constructed of capsules and spheres are used to animate the hand movements in VR. This setup is illustrated in Figure 4.7.

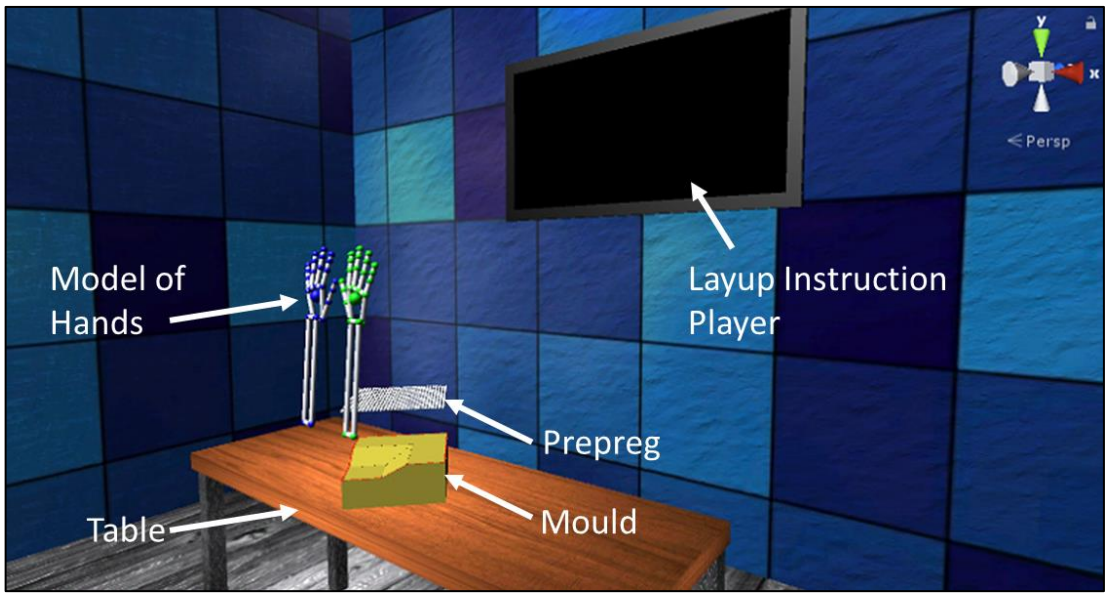


Figure 4.7 - Schematic of the virtual reality training simulation environment

4.3.2 Simulation Mechanics

This VR Training simulation is designed as a video game with multiple stages. The user must achieve specific goals at each stage as shown in the layup instruction player in order to advance to the next stage. Key components that make up the simulation mechanics are explained below:

- **Prepreg deformation:** No kinematic modelling was used in modelling the deformation of the prepreg. All deformations of the prepreg programmed in to the VR simulation are pure animations to match as closely as possible to the real

world deformations of the prepreg upon performing physical manipulations to it.

- **Interaction of the prepreg with the mould:** The only form of interaction of the prepreg with the mould programmed in to the simulation is individual spheres making up the net coming in to contact with the mould. To detect such gain of contact, Unity 3D's inbuilt collider components are used. A sphere collider is attached to each node of the net and appropriate colliders are attached to the mould at the design stage of each layup action (use of colliders will be explained in more detail in section 4.3.3)
- **Interaction of hands with the prepreg:** The user is given control over the position, orientation and state of the prepreg based on the interaction state between hand models and the prepreg model. This interaction state is defined by the interaction between appropriate colliders attached to hand models and spheres making up the prepreg at each action stage. This will become clearer in section 4.3.3.
- **The layup instruction player:** This particular component had to be constructed upon the completion of the entire VR simulation. The completed VR simulation was played through by the author as it was intended to be played, while the gameplay being recorded. The recorded gameplay was then trimmed in to respective action stages and integrated in to the simulation to be played at each action stage on the layup instruction player as guidance for new users.

4.3.3 Design of the Action Sequence

In this section, the design of each action of the VR training simulation as listed in Table 4.1 will be discussed in detail. A video recording of the entire VR simulation being played by the author is provided in [128]. In order to better understand this section of this chapter, it is advised to view this video. It is important to understand one key design principle that has been used throughout the simulation before this discussion. While the Leap Motion controller is capable of tracking the position and orientation of hands and fingers in real time, it was experienced that there is a significant amount of noise in the

Leap Motion controller's interpretation of finger joint orientations. To address this issue, once the user has achieved a certain required hand gesture within the simulation, real time finger tracking is no longer displayed to the user during the VR simulation. Instead, the user is shown pre designed hand models of both hands within the simulation, while their global positions and orientations being updated with the Leap Motion controller's data. This concept will be clearer within the discussion of the design of Action 1 below.

Action 1: First step in action 1 is to pick up the prepreg. Figure 4.8 illustrates the starting view of the user in action 1. The user is shown two predesigned hand models in pink and white colours that are holding the prepreg. A box collider (as illustrated in green wire frames) each is attached to these hands. At this point, the video displayed on the layup instruction player is instructing the user to move his/her hands to match the position and orientation of the predesigned hand models. Once the user has followed these instructions, he/she will gain control over the position and orientation of the prepreg. The simulation identifies that the user meets these instructions once the colliders attached to the tip of the thumb and middle fingers of the left and right hands as illustrated in Figure 4.9 are simultaneously intersecting with the box colliders attached to the left and right predesigned hand models respectively. A similar mechanism is used in the design of all other action stages in trying to get the user to match his/her hands with the predesigned hand models unless otherwise stated.

It is important to identify the coordinate system of the simulation at this point in order to understand how the user's hand positions and orientations are translated in to that of the prepreg after picking up the prepreg. With reference to Figure 4.8, positive X direction is from left to right, positive Y direction is from bottom to top and positive Z direction is in to this figure. With this coordinate system in mind, once the user has picked up the prepreg, its position and orientation in 3D space is controlled by the user by the following criteria:

- In reference to Equation 4.1, palm positions are acquired by the Leap Motion controller at each frame. The X, Y and Z positions of the prepreg is then updated as an average of the two palm positions at each frame.

$$Prepreg\ Position\ [X,Y,Z] = (Left\ Palm\ Position + Right\ Palm\ Position\ )/2 \quad \textbf{Equation 4.1}$$

- Angle of rotation of the prepreg about the X-axis (X_rot), with the positive X_rot direction defined by the right hand rule with the thumb pointing towards the positive X direction is calculated using the position of the mid-point of the Pinky (little) finger relative to the position of the wrist. The little finger was chosen here due to the Leap Motion controller having the best line of sight to it compared to other fingers. The formula used to calculate X_rot is illustrated in Equation 4.2. Here, RWP = Right Wrist Position, RPP = Right Pinky Position, LWP = Left Wrist Position and LPP = Left Pinky Position. As evident from this formula, X_rot is an average of the orientations of both the left and right hands.

$$X\_rot = [\ (Tan^{-1}((RPP[Y] - RWP[Y])\ /\ (RPP[Z] - RWP[Z]))) + (Tan^{-1}((LPP[Y] - LWP[Y])\ /\ (LPP[Z] - LWP[Z])))]/2 \quad \textbf{Equation 4.2}$$

- Angle of rotation of the prepreg about the Y-axis (Y_rot) and Z-axis (Z_rot), using the same right-hand rule as explained above are calculated using the relative positions between left and right palms. Palm positions were used in this case due to the Leap Motion controller having the best line of sight to them compared to fingers and also the least amount of noise in terms of real time tracking. The formulae to calculate Y_rot and Z_rot is provided below:

$$Y\_rot = Tan^{-1}((RPP\ [X] - LPP\ [X])/\ (LPP\ [Z] - RPP\ [Z])) \quad \textbf{Equation 4.3}$$

$$Z\_rot = Tan^{-1}((LPP\ [Y] - RPP\ [Y])/\ (RPP\ [X] - LPP\ [X])) \quad \textbf{Equation 4.4}$$

Here, RPP = Right Palm Position and LPP = Left Palm Position.

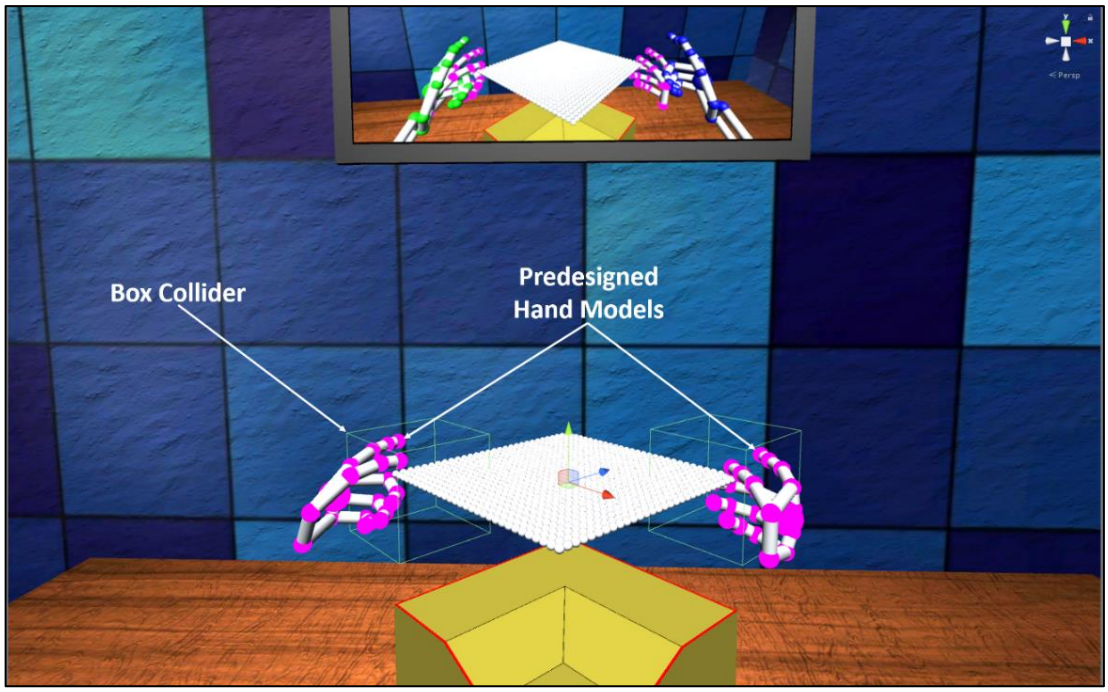


Figure 4.8 – Starting view of the user in action 1

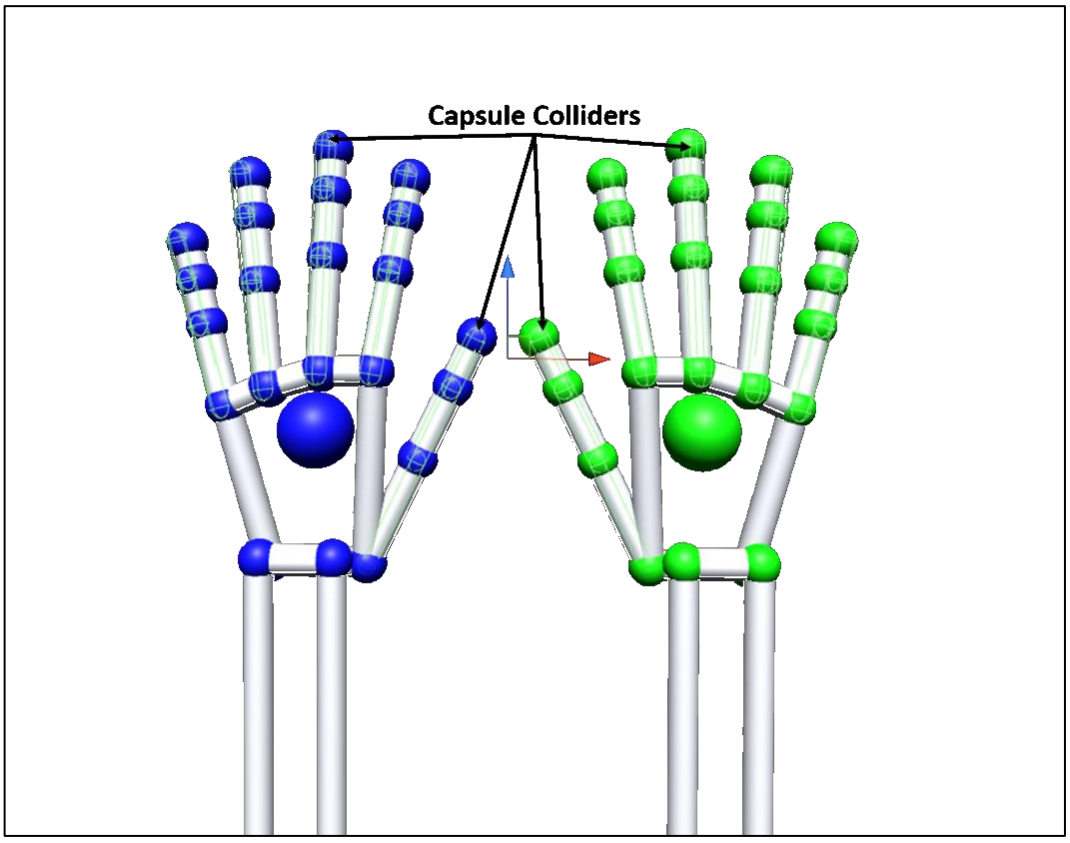


Figure 4.9 – Hand models updated with Leap Motion controller data

Once the prepreg has been picked up by the user, the final step in action 1 is to place it on the mould such that the prepreg meets the datum marked on the mould. In order to achieve this, three sphere colliders are placed at the top three corners of the mould as illustrated in Figure 4.10. When the sphere colliders attached to the corresponding three nodes of the prepreg intersect with the sphere colliders attached to the mould, action 1 is completed.

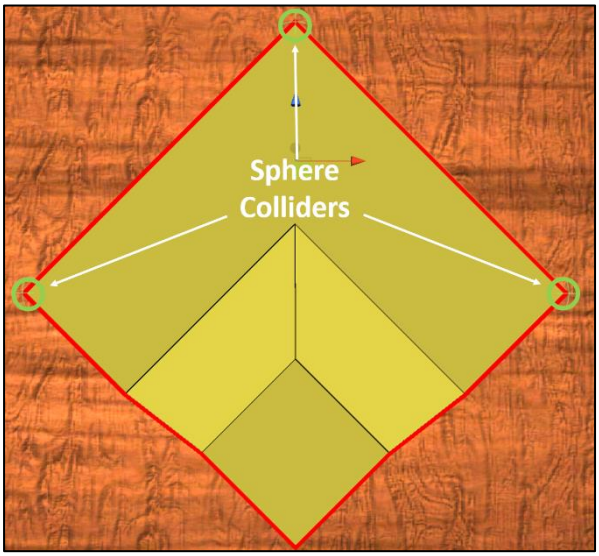


Figure 4.10 - Illustration of sphere colliders attached to the mould to detect placement of prepreg

Action 2: Once the user gets the hands in position with the predesigned hand models, the right hand model is kept fixed and the left hand model's position in 3D space is linked directly to the palm position of the left hand from the Leap Motion controller. A sphere collider is attached to the tip of the thumb of the left hand model as illustrated in Figure 4.11 and the user is instructed to secure the prepreg on to Edge 1. When the sphere collider attached at the thumb tip comes in to contact with the sphere colliders attached to the nodes of the prepreg at the corresponding edge, the nodes change colour from white to black indicating that they have been secured on to the mould. Once all nodes at the edge have been secured as explained, action 2 is completed.

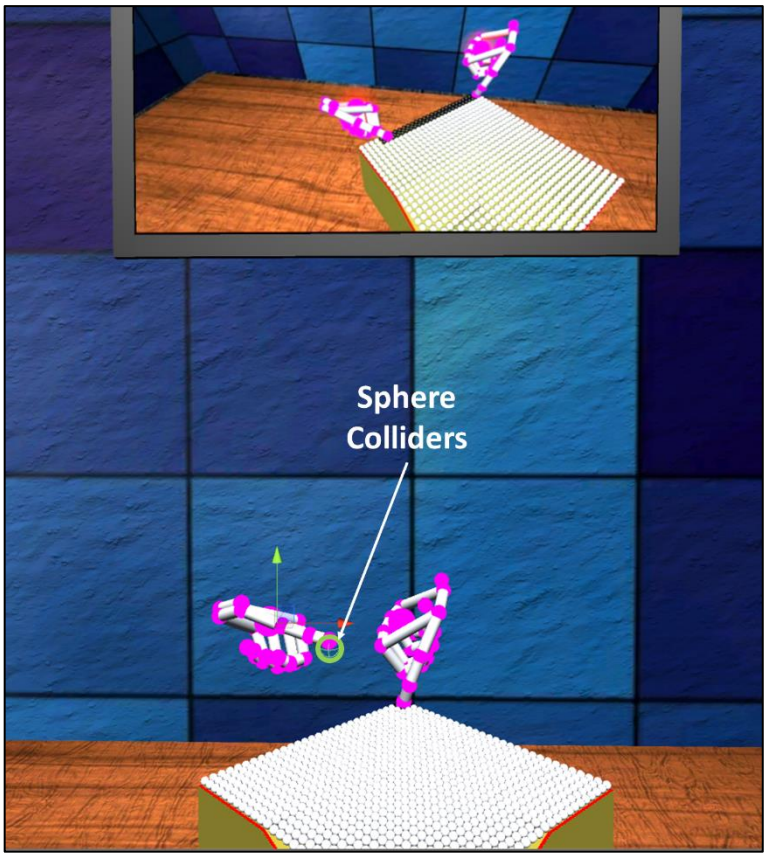


Figure 4.11 - Starting view of the user in action 2

Action 3: Design methodology of action 3 is identical to that of action 2. Only difference is that the left hand model is kept fixed while the right hand model is used to secure the prepreg on to Edge 2.

Action 4: This action includes using the Dibber to secure the prepreg on to section A. The left hand model is kept fixed during this action stage. Two colliders are attached to the Dibber (red colour tool) as illustrated in Figure 4.12. Once the collider attached to the tip of the thumb of the right hand model intersects with the sphere collider attached to the Dibber, the user gains control over the position of the Dibber in 3D space. The position of the Dibber and the predesigned right hand model together in 3D space is linked directly to the palm position of the right hand from the Leap Motion controller. Their orientation is kept fixed. Once the box collider attached to the Dibber comes in to contact with any of the sphere colliders attached to the nodes of the prepreg within section A, these nodes change colour from white to black indicating they have been

secured (cannot be moved) on to the mould. Once all nodes in section A have been secured to the mould, action 4 is completed.

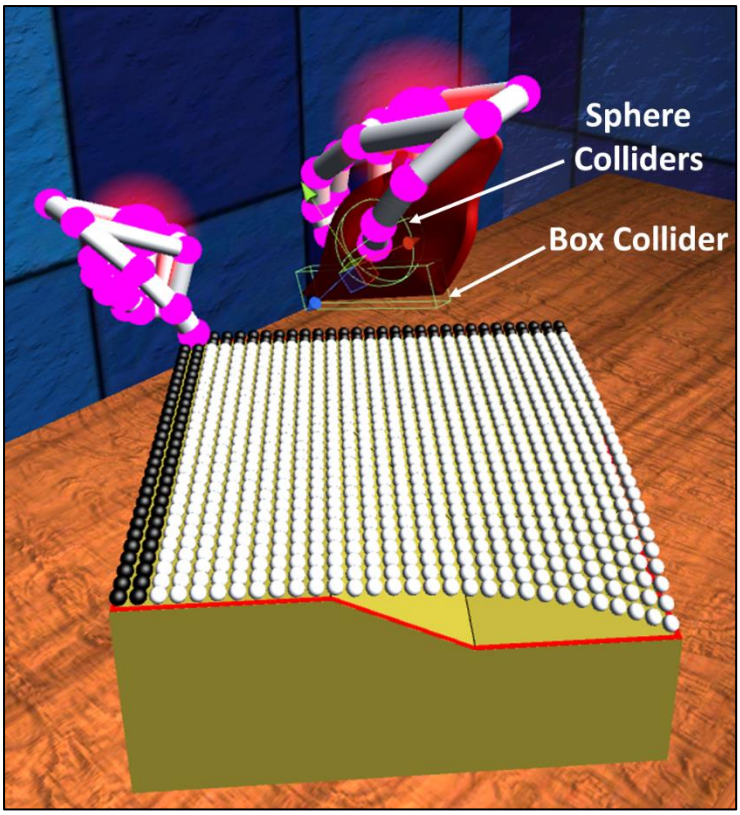


Figure 4.12 - Starting view of the user in action 4

Action 5: Design methodology used for action 5 is identical to that of action 4. Only difference is that the user is supposed to secure all the nodes of the prepreg in section B on to the mould.

Action 6: This action includes securing the prepreg to Edge 3 using the left hand. Predesigned model of the right hand is kept fixed throughout this action stage. The left hand is linked directly to the palm position of the left hand from the Leap Motion controller. A sphere collider is attached to the tip of the thumb of the left hand as illustrated in Figure 4.13. Once this sphere collider intersects with sphere colliders attached to the nodes of the prepreg lying on Edge 3, they change colour from white to black indicating to the user that they have been secured to the edge. Once all nodes on Edge 3 have been secured to Edge 3, action 6 is completed.

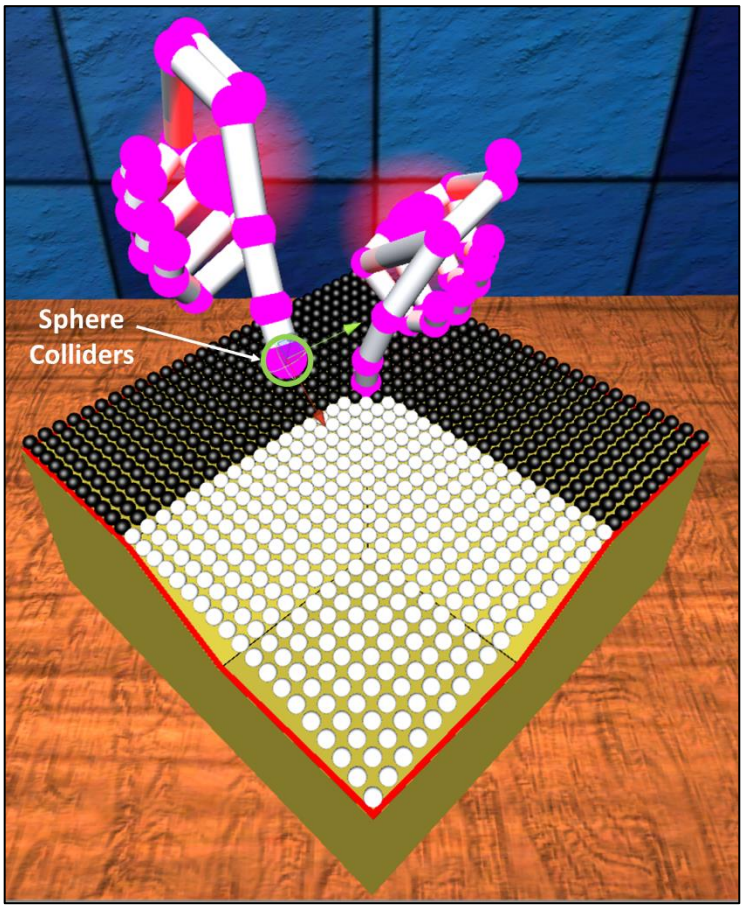


Figure 4.13 - Starting view of the user in action 6

Action 7: Design methodology used for action 7 is identical to that of action 6. Only difference is that the left hand model is kept fixed while the right hand model is used to secure the prepreg on to Edge 4.

Action 8: This action includes securing the prepreg in to the section of Groove 1 enclosed by sections C and D. A sphere collider is attached to the tip of the middle finger of the right hand model as illustrated in Figure 4.14. Two box colliders are attached parallel to the two slanted faces of the mould made up of sections C, E, and G and D, F, and H. Three diagonal nodes, as illustrated in Figure 4.15 must be pushed in to Groove 1 until the sphere colliders attached to them come in to contact with the box colliders attached to the slanted faces. This push down movement of nodes is achieved by making their Y position equal to the Y position of the sphere collider attached to the tip of the thumb of the right hand model while the colliders are intersecting with each other. The Y position of the nodes are also clamped to ensure that they do not move in the upward

direction or below the surface of the mould. When these diagonal nodes come in to contact with the box colliders, their colour is changed from white to black to indicate they have been secured to the mould. Once all three nodes make contact with the mould, action 8 is completed.

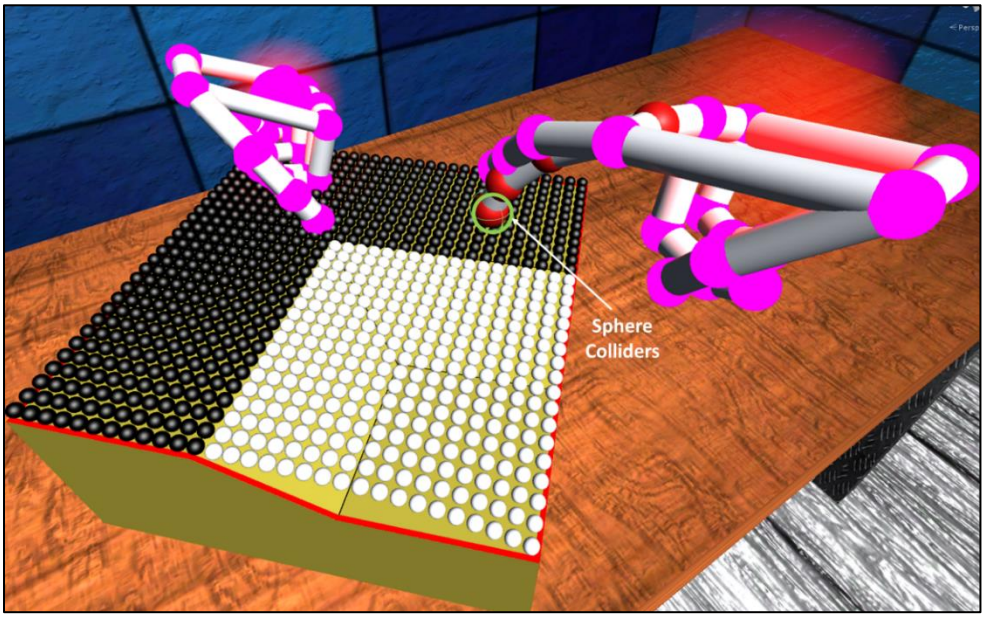


Figure 4.14 - Starting view of the user in action 8

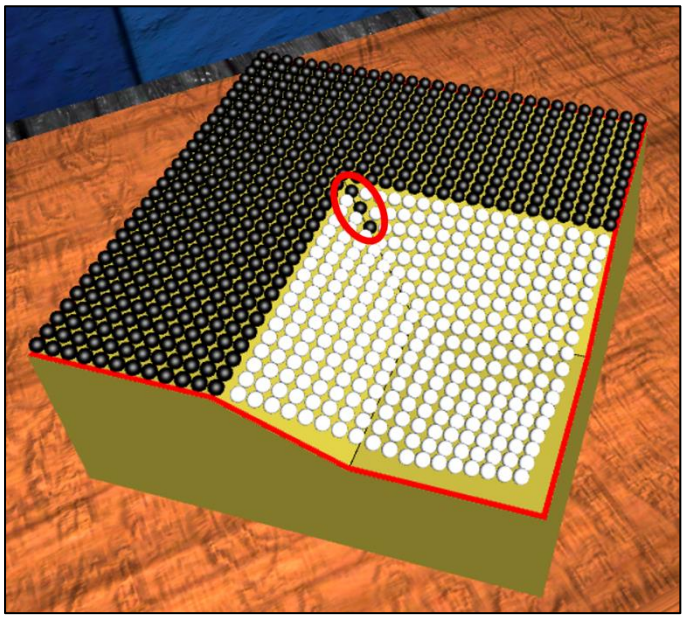


Figure 4.15 – State of the prepreg at the end of action 8

Action 9: This action includes securing the prepreg on to section C using the Dibber. Design methodology in giving the user control over the Dibber is identical to that of action 4. To complete this action, the user must move the three rows of nodes in line with the three diagonal nodes that were pushed down in action 8 as highlighted by a red ellipse in Figure 4.16. This push down movement of nodes is achieved by making their Y position equal to the Y position of the box collider attached to the Dibber while the colliders are intersecting with each other. The Y position of nodes are clamped in a similar manner to that in action 8 to stop them from moving below the mould surface. Once the user brings all three rows of nodes in to contact with the slanted face of the mould below them, this action is completed.

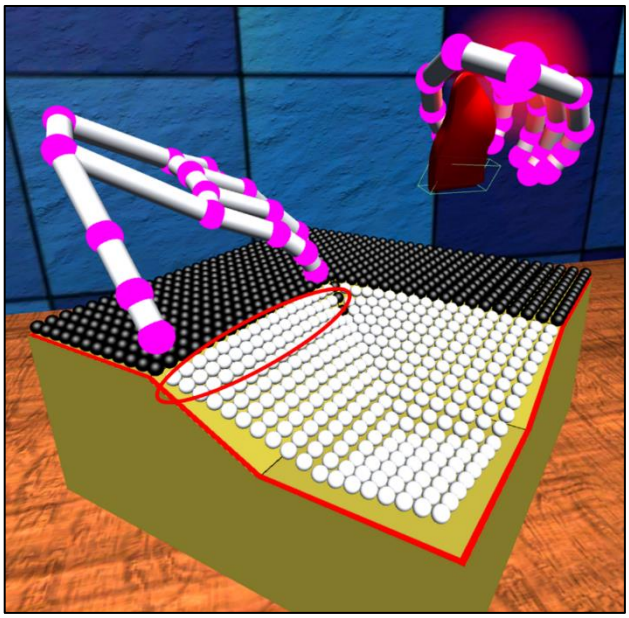


Figure 4.16 - Starting view of the user in action 9

Action 10: Design methodology used for action 10 is identical to that of action 9. Only difference is that the user is supposed to push down three rows of nodes that come in to contact with section D of the mould. State of the prepreg at the end of action 10 is illustrated in Figure 4.17.

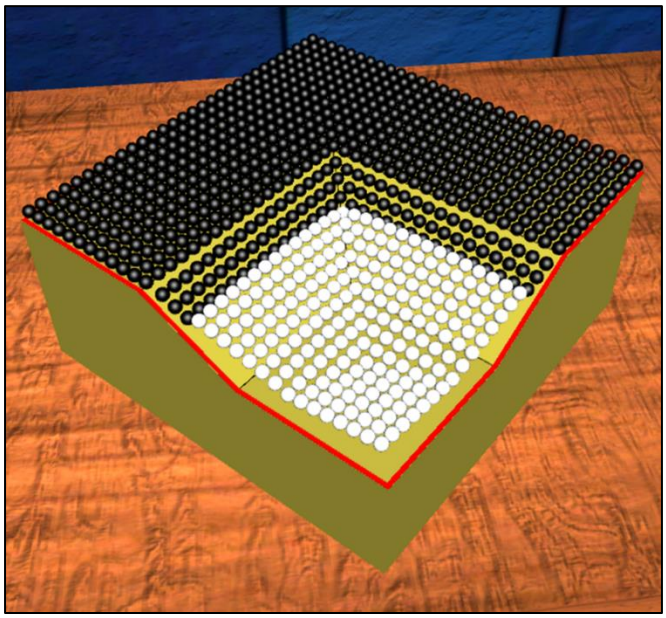


Figure 4.17 - State of the prepreg at the end of action 10

Action 11: Design methodology used for this action is identical to that of action 8. The only difference is that the user is supposed to push down the three diagonal nodes that come in to contact with the section of Groove 1 enclosed by sections E and F. State of the prepreg at the end of action 11 is illustrated in Figure 4.18.

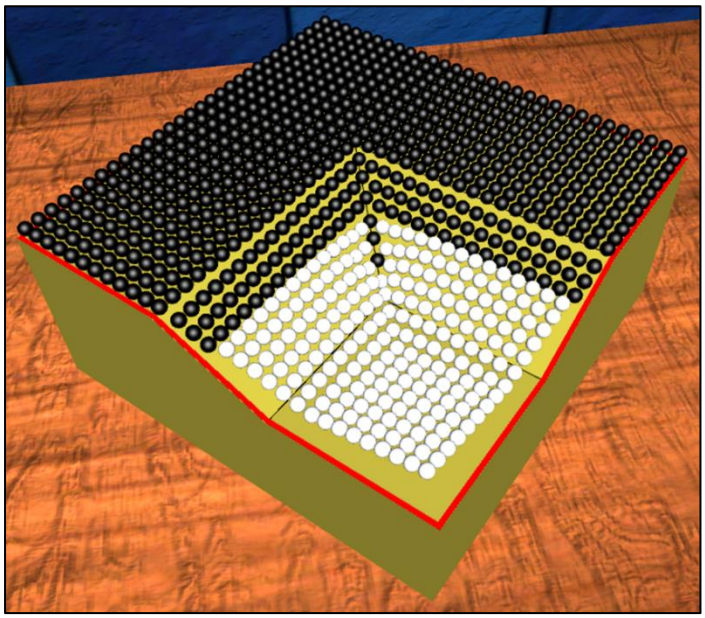


Figure 4.18 - State of the prepreg at the end of action 11

Actions 12 and 13: Design methodology used for these 2 actions is identical to that used for action 9. The user is supposed to push down the rows of nodes that come in to contact with sections E and F of the mould. State of the prepreg at the end of action 13 is illustrated in Figure 4.19.

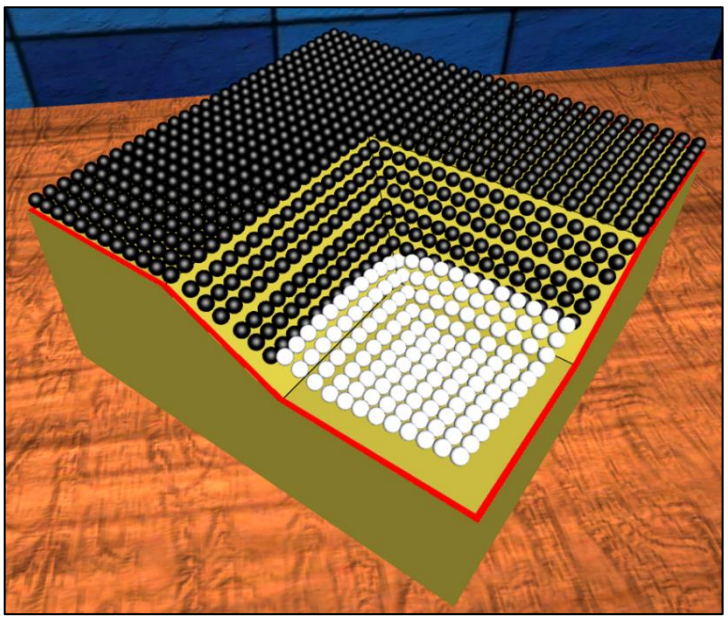


Figure 4.19 - State of the prepreg at the end of action 13

Action 14: Design methodology used for this action is identical to that of action 8. The only difference is that the user is supposed to push down the two diagonal nodes that come in to contact with the section of Groove 1 enclosed by sections G and H. State of the prepreg at the end of action 14 is illustrated in Figure 4.20.

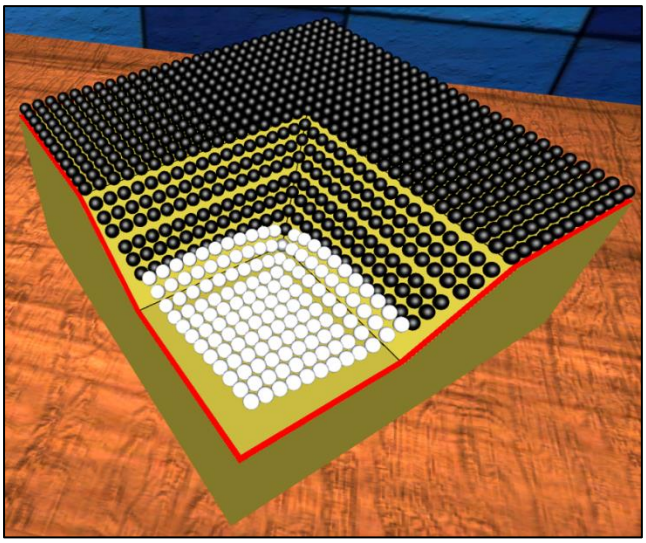


Figure 4.20 - State of the prepreg at the end of action 14

Actions 15 and 16: Design methodology used for these two actions is identical to that used for action 9. The user is supposed to push down the rows of nodes that come in to contact with sections G and H of the mould. State of the prepreg at the end of action 16 is illustrated in Figure 4.21.

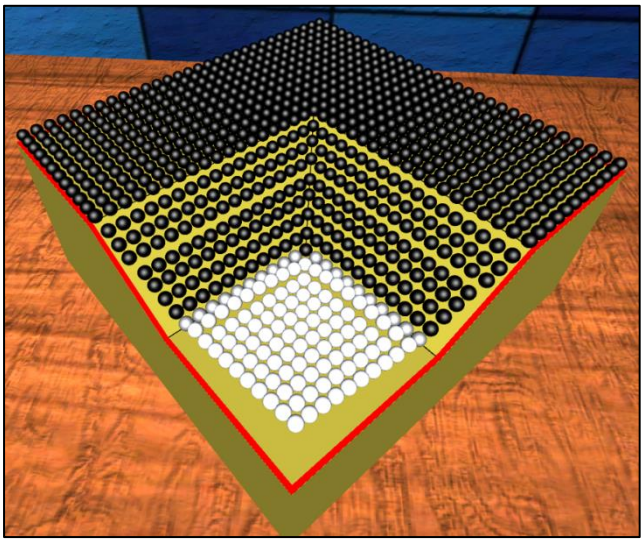


Figure 4.21 - State of the prepreg at the end of action 16

Actions 17: This action includes securing the prepreg in to Groove 2. Left hand model is kept fixed with the thumb at the intersection between Grooves 2 and 3. A sphere collider is attached to the tip of the middle finger of the right hand model as illustrated in Figure 4.22. Once this sphere collider intersects with sphere colliders attached to the nodes of

the prepreg lying in Groove 2, they change colour from white to black indicating to the user that they have been secured in to the groove. Once all nodes in Groove 2 have been secured, action 17 is completed.

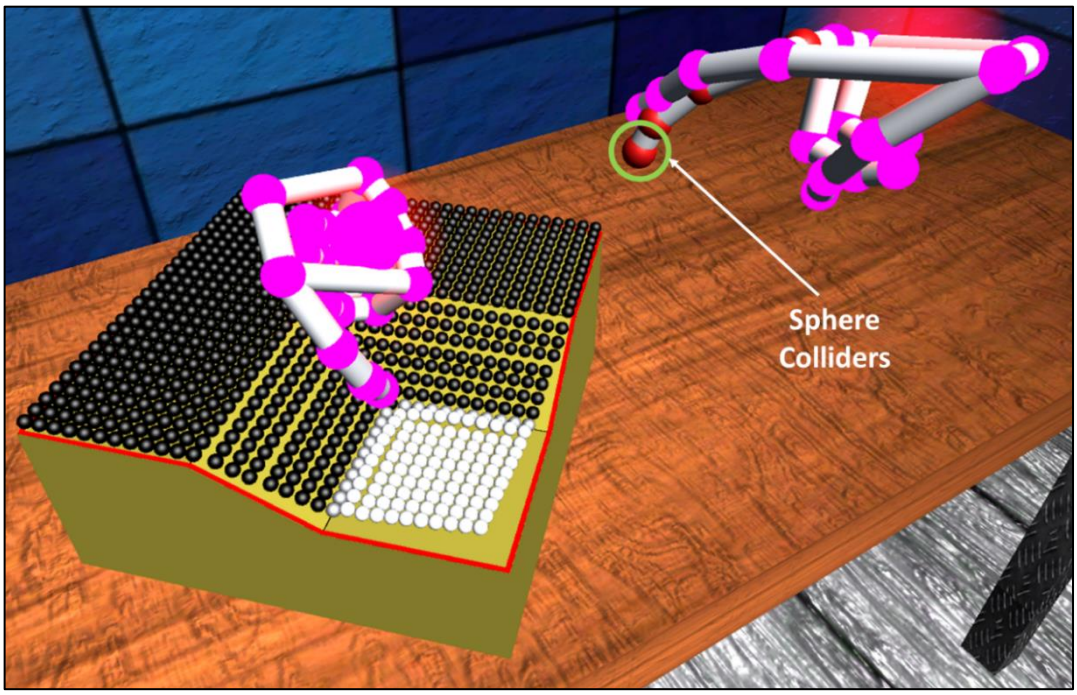


Figure 4.22 - Starting view of the user in action 17

Action 18: Design methodology for this action is identical to that of action 17. The user is supposed to secure the nodes lying in Groove 3 instead of Groove 2. Once, all these nodes have been secured by the user, action 18 is completed.

Action 19: This action includes securing the prepreg on to section I using the Dibber. Design methodology for this action is identical to that of action 4. However, there is no function set for the left hand in this action. Once the user ensures that the box collider attached to the Dibber as illustrated in Figure 4.23 gains contact with all sphere colliders attached to nodes lying on section I of the mould, they change colour from white to black indicating they have been secured on to the mould, thus completing action 19.

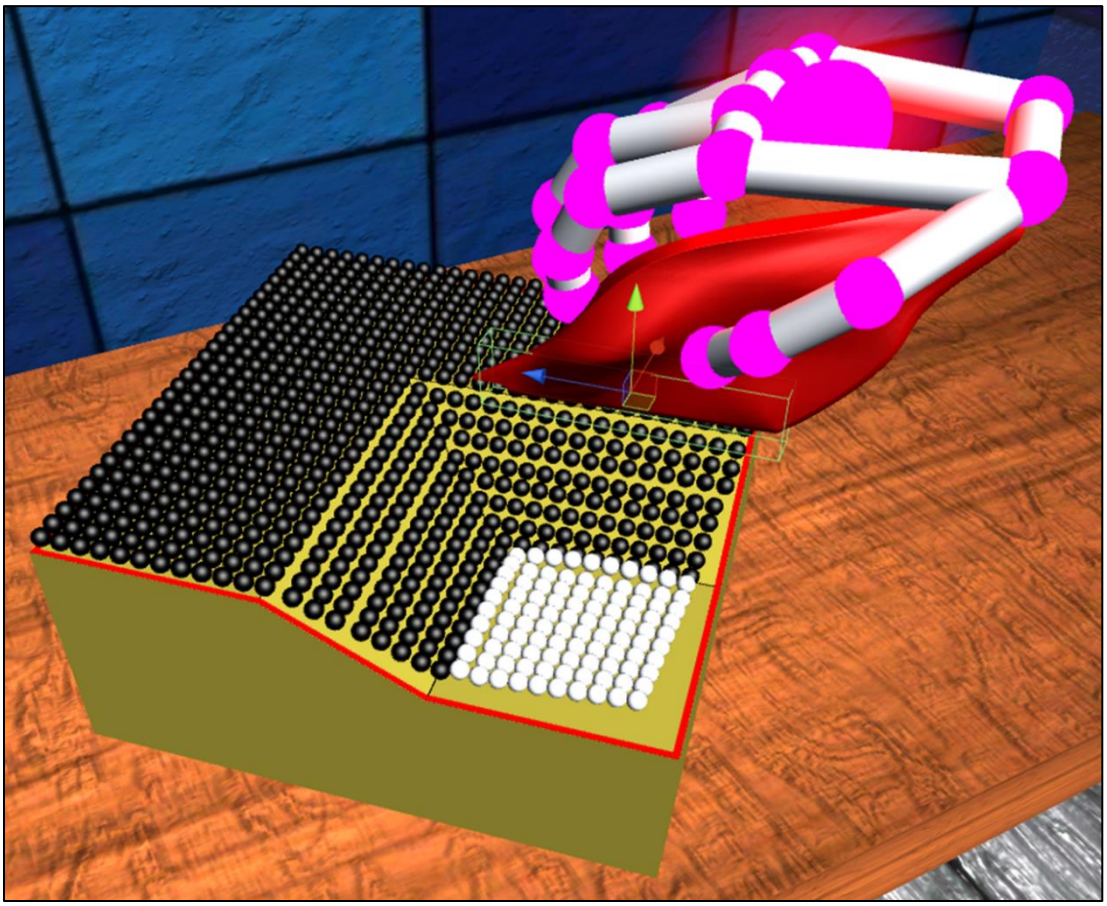


Figure 4.23 - Starting view of the user in action 19

4.4 Experimental Methodology

An experiment was carried out to compare the effectiveness of the VR training simulation against one to one layup training. Two groups made up of 10 novice laminators each were used for this experiment.

One group was made to play through the designed VR training simulation 3 times with an experimental setup as shown in Figure 4.24. The infrared sensor was attached to a tripod and its height was adjusted for each candidate to calibrate the VR training simulation with the height of each candidate. Each candidate was video recorded during their VR training and screen recordings of their attempts at the VR training simulation were also collected. Prior to the candidates playing through the VR training simulation, they were provided with the following instructions:

- Explanation of the working mechanics of the Oculus Rift and the Leap Motion controller: Especially the fact that there is a limited field of view for the Leap Motion controller and that if hand tracking failed during the simulation they must bring their hands back within the field of view of the Leap Motion controller.

- The fact that the VR simulation is analogous to a video game that requires following instructions played on the layup instruction player in order to advance to the end of the simulation.

- The fact that predesigned hand models are used throughout the VR simulation in order to address the inability of the Leap Motion controller to maintain accurate finger tracking. This was demonstrated to the candidates by the author by performing action 1 of the simulation.

Immediately after the candidates have completed playing through the VR training simulation, they were made to layup the same exact mould under clean room conditions while being video recorded by a camera facing downwards on the mould.

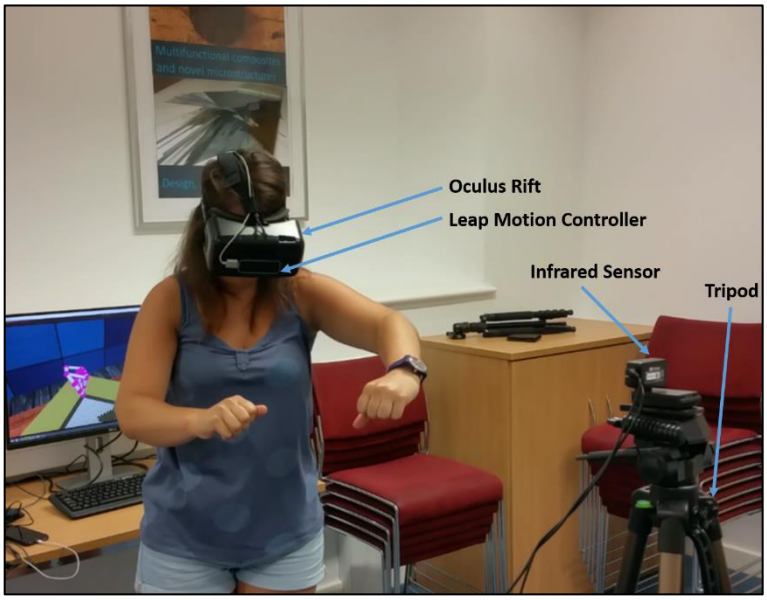


Figure 4.24 - Experimental setup for virtual reality training

(Permission has been granted from the test candidate to use Figure 4.24 as part of this thesis)

The second group was provided with one to one training by the author, where the candidates were made to observe the author performing the layup sequence as described in section 4.2.2 three times in a row. Immediately after one to one training, candidates were made to layup the same exact mould under clean room conditions while being video recorded by a camera facing downwards on the mould. This was followed by the candidates playing through the VR simulation two times and filling up the questionnaire illustrated in Figure 4.25. This group of candidates was also provided with the same set of instructions that was provided to the previous group.

The objective of this questionnaire was to get preliminary feedback from the candidates on how the current design of the VR training simulation can be improved to provide a better learning experience. Questions 1 and 2 of the questionnaire were answered using a 5-point Likert Scale [129].

VR Training Simulation Effectiveness Questionnaire

[1-Strongly Disagree, 2-Disagree, 3-Neither Agree nor Disagree, 4-Agree, 5-Strongly Agree]

Question 1: Having the Virtual Reality training simulation prior to my first layup would have been useful (Rate from 1-5)

.....

Question 2: The Virtual Reality training aid could act as a standalone training aid. I.e. one to one training is unnecessary (rate from 1-5)

.....

Question 3: Which method of training would you prefer? (Tick one box)

i) One to one training ☐
ii) Virtual Reality training ☐
iii) One to one training + Virtual Reality training ☐

Question 4: What needs to be changed or added/removed in to the current Virtual Reality training aid to provide a better training experience?

Figure 4.25 – Virtual reality training simulation effectiveness questionnaire

4.5 Experimental Results and Discussion

4.5.1 Time to Completion of the VR Training Simulation

Time to completion of each attempt at the VR training simulation for all 20 candidates from the two groups was calculated using the screen recordings of the VR training simulations. Starting point of each attempt was noted when the candidate picks up the virtual prepreg within the simulation and the ending point when the user completes action 19. Percentage reduction in time from the first to last attempt of each candidate was also calculated. This can be used as an indication of how easily the candidates got used to the working mechanics of the VR training simulation. Results for the first group who played through the VR simulation prior to laying up in the clean room are illustrated in Table 4.2. Results for the remaining group are illustrated in Table 4.3.

Results from Table 4.2 indicate that there is a positive percentage reduction in time to completion for every candidate. Furthermore, there is a significant (>50%) average percentage reduction in time to completion, indicating that the candidates' familiarity with the working mechanics of the VR simulation has significantly improved with just three attempts at it. It must also be noted that all candidates who took part in this experiment had no previous experience of interacting with VR systems.

Similar to the first group, every candidate in the second group has also shown a positive percentage reduction in time to completion as evident from Table 4.3. It can also be noted that the average of time to completion of attempt 1 for the second group is 128.3 s lower than that of the first group. This value indicates a percentage difference of 54.4% relative to the average time to completion of attempt 1 of the second group. This significance difference must be caused because candidates of the second group went through one to one training by the author followed by them laying up the actual mould in the clean room prior to playing through the VR simulation. Therefore, candidates of the second group had a much better mental representation of the steps required to complete the layup when compared to the candidates of the first group. Further statistical tests are not performed on these time metrics due to the difference in the training conditions between the 2 groups.

Table 4.2 – Time to completion of the VR training simulation results for the first group (TTC – time to completion, % Reduction in Time = [(TTC Attempt 1 – TTC Attempt 3)/TTC Attempt 1]*100)

Candidate	TTC – Attempt 1 (seconds)	TTC – Attempt 2 (seconds)	TTC – Attempt 3 (seconds)	% Reduction in Time
1	357	210	133	62.7
2	347	203	158	54.4
3	478	337	177	63.0
4	295	208	130	55.9
5	374	241	232	38.0
6	365	209	173	52.6
7	182	157	124	31.9
8	551	312	196	64.4
9	317	226	186	41.3
10	375	236	185	50.7
Average	**364.1**	**233.9**	**169.4**	**53.4**

Table 4.3 - Time to completion of the VR training simulation results for the second group (TTC – time to completion, % Reduction in Time = [(TTC Attempt 1 – TTC Attempt 2)/TTC Attempt 1]*100)

Candidate	TTC – Attempt 1 (seconds)	TTC – Attempt 2 (seconds)	% Reduction in Time
1	155	121	21.9
2	198	148	25.3
3	243	198	18.5
4	229	150	34.5
5	278	141	49.3
6	242	155	36.0
7	304	183	39.8
8	222	82	63.1
9	282	187	33.7
10	205	130	36.6
Average	**235.8**	**149.5**	**36.6**

4.5.2 VR Training Simulation Effectiveness Questionnaire

Responses from all ten candidates of the second group to the VR training simulation effectiveness questionnaire are illustrated in Table 4.4.

Table 4.4 – Results from the VR training simulation effectiveness questionnaire (Refer to Figure 4.25)

Candidate	Q1	Q2	Q3	Q4
1	5	5	ii	- Add voice feedback or indicate on the layup instruction player when the user is not performing the correct movement. - Equipment used could be made more comfortable, especially in terms of having fewer wires. - Having no vision of the real world makes movements tricky.
2	4	4	iii	- Provide more training at the beginning to better understand the working mechanics of the simulation. - The jumps after each successful action is a bit disorienting. - Impossible to include the issue of tackiness of prepreg.
3	3	2	iii	- Having a real mould would have been helpful to make it feel more real.
4	4	5	iii	- Provide audio instructions.

Candidate	Q1	Q2	Q3	Q4
				- Add a pause between actions, so that the user can see more clearly that an action has been completed.
5	3	4	iii	- VR training in thin air is not interactive and does not create muscle memory. Use a real mould to make it more realistic. - Provide audio instructions.
6	4	2	iii	- Indicate how much strength/pressure that must be applied to the prepreg to complete the layup.
7	2	2	iii	- Provide haptic feedback. - Provide the ability to ask questions or get tips within the VR simulation.
8	4	4	iii	- Provide physical feedback (use a real mould). - Lower the layup instruction player so that it stays visible during layup. - Provide audio instructions on why each action is performed in the way it is instructed.
9	4	4	iii	- Incorporate more physical things, such as a real mould in to the VR training simulation.
10	5	1	iii	- Provide instructions that explain in detail how to achieve good quality during the layup
Average	**3.8**	**3.3**	**-**	**-**

(Table 4.4 continued)

Results from Table 4.4 indicate that Question 1 received an average answer of 3.8, which means that the group in general agree that having such a VR training simulation prior to their first layup attempt would have been useful. However, Question 2 received an average answer of 3.3, which suggests that the group is reluctant to agree on the notion that the VR training simulation can act as a standalone training aid in its current form. This is further confirmed by the responses received for question 3, where 9 out of 10 candidates ticked on option iii, that suggest that they feel both one to one training and VR training is required for a complete training experience.

The answers provided by the candidates for Question 4 act as valuable feedback for improving the effectiveness of a second iteration of the current VR training simulation. The candidates through these answers have pointed out major drawbacks of the current VR training simulation. These drawbacks are discussed below.

- Lack of some form of instructions, further explaining how to accurately take actions to ensure that the quality of the layup is maintained. The candidates, several times in their answers have also mentioned a possible solution to this drawback in the form of audio instructions for each action. This can easily be implemented in a second iteration.
- The candidates have also pointed out the lack of haptic feedback in the VR simulation several times. As discussed in Chapter 2 of this thesis, there are several options to provide haptic feedback. These include haptic gloves and exoskeletons. Current haptic gloves use a fixed number of vibrating motors placed at certain points on the gloves to emulate force feedback. At this point, this doesn't seem to be the solution to emulating the force required to deform a prepreg. In comparison, an exoskeleton might do a better job in simulating this phenomenon. However, conclusions cannot be made on this without carrying out experiments using both technologies. Candidates have also requested for the presence of a real mould during the VR training simulation as a solution to this problem. The author agrees that this a valid suggestion and can certainly be implemented in a second iteration. However, the presence of just a mould during the VR simulation will only provide a portion of the

total haptic feedback involved with the layup process and this could lead to unforeseen confusions to the user and complications during the design stage. Any further conclusions on such a use of a mould cannot be made without actually implementing it and carrying out experiments on it.

Other points made by the candidates are briefly discussed below.

- Having fewer wires with equipment involved: While the amount of wires used was kept at a minimum at the point of design of this VR simulation, novel wireless solutions for VR headsets have been recently introduced [130] and these novel solutions can be used in a second iteration to further improve the comfort of using the Oculus Rift.
- Access to vision of the real world: This can be achieved by using an appropriate Augmented Reality headset instead of the Virtual Reality headset.
- Including tackiness and force-displacement properties of the prepreg: This is not possible as a VR solution. Achieving this would require the development of a novel reusable material capable of mimicking the physical properties of a prepreg material.

4.5.3 Accuracy and Time to Completion of the Clean Room Layup

All 20 candidates from the two groups were scored on accuracy of their clean room layup through video analysis. The scoring criteria used for actions 1-19 is summarised in Appendix: Table 4. Every criterion under which each action is scored can receive a score of 1 or 0, where 1 = successful and 0 = failure, based on the judgement of the author.

Results from the above accuracy study and time to completion of the layup for all 20 candidates from the two groups are illustrated in Table 4.5. The maximum score a candidate can receive is 53.

Table 4.5 – Results from the layup accuracy study and time to completion (TTC) of the layup for groups 1 and 2

Candidate	Group 1		Group 2	
	Accuracy Score	TTC (seconds)	Accuracy Score	TTC (seconds)
1	41	176	26	108
2	37	241	36	145
3	43	580	23	205
4	38	250	40	130
5	39	270	32	91
6	23	280	28	280
7	33	190	41	100
8	27	441	26	138
9	40	262	27	58
10	36	358	12	230
Average	**35.7**	**304.8**	**29.1**	**148.5**

Results from Table 4.5 indicate that group 1 has a higher average accuracy score compared to that of group 2. There is a percentage difference of 22.7% between these two values relative to the average accuracy score of group 2. Through an independent T-Test, it can be shown that Group 1 (Mean = 35.7, SD = 6.3), who received VR training prior to their layup in the clean room, compared to Group 2 (Mean = 29.1, SD = 8.6), who received 1 to 1 training, paid more attention to ensuring that they followed the provided layup instructions accurately, $t(18) = 1.95$, $p=0.067$.

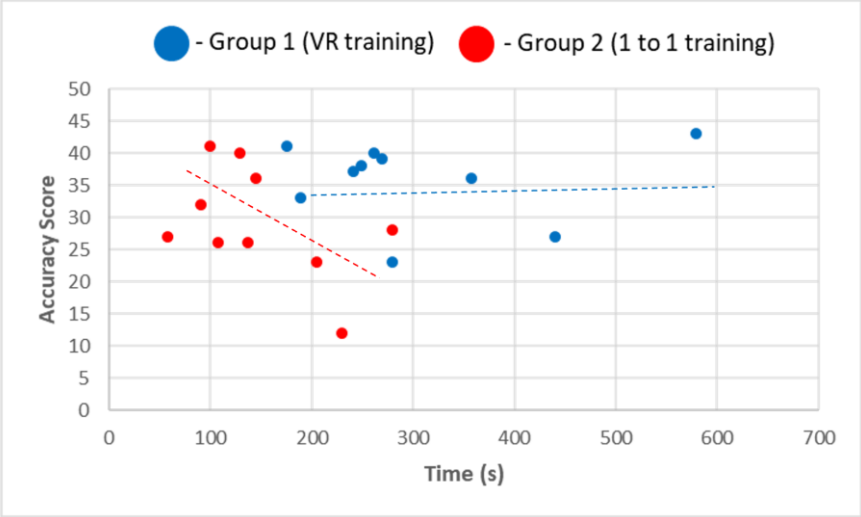


Figure 4.26 - Accuracy - time plot for all candidates of groups 1 and 2

Interesting conclusions can also be made by analysing a time-accuracy plot as illustrated in Figure 4.26. Group 2, that went through 1 to 1 training seem to show a decrease in accuracy with time spent. This could be an indication of different levels of capabilities individuals have in terms of draping prepreg material. Amount of time spent then becomes inversely proportional to the skill level of an individual. However, the same trend cannot be observed with group 1. In general, group 1, that went through VR training does not seem to show a trend of decreasing accuracy with time spent.

In terms of time spent on individual actions by candidates, on average, the most amount of time (30.1 s) was spent on Action 1 trying to align the prepreg with the datums on the mould and readjusting it during the layup to ensure that the prepreg didn't slide away from the datums.

Comparing the average time to completion of the two groups indicate that there is a percentage difference of 105.3% between these two values relative to the average time to completion of group 2. This means that the candidates who received VR training first took a significantly longer time to complete the layup than the candidates who received one to one training prior to their clean room layup. There could be three reasons for this significant difference in time:

1. Candidates from group 2, having already seen the author perform the layup three times in front of them found it easier to quickly reproduce the layup.
2. Candidates from group 1 did not get the opportunity to observe the layup of the real mould prior to their attempt at it. Therefore, unlike the candidates from group 1, candidates from group 2 did not have a clear mental representation of the layup environment.
3. Candidates from group 2, having played through the VR training simulation 3 times prior to their attempt at the clean room layup were more conscious about accurately following the layup instructions. This led to them taking more time to ensure that their hand gestures and grip styles matched exactly to that of the VR training simulation prior to performing each action. This is further confirmed by their higher average accuracy score as previously discussed.

4.6 Summary

Use of VR technology as a platform for training novice laminators has been explored in this chapter. The case study presented in this chapter is no more than a preliminary study that provides an idea about whether VR technology is suitable or not as a platform for hand lamination training. Overall feedback received by the test candidates suggest that the VR training aid in its current form may not be suitable as a standalone platform for hand lamination training, instead it is more suitable to be used in tandem with one-to-one training. However, experimental results did not fully agree with the opinions of the test candidates, since test candidates who received VR training only performed better (>20% increase in average accuracy score) than the test candidates who received one to one training during their layup trials. In addition to this, there was a >50% reduction in average time to completion of the VR training aid just after 3 trials at it. This means that even first time users of such a VR training aid can quickly grasp its working mechanics. However, it should be noted that similar results may not have been obtained if a more complex mould shape that required complex manipulations to the prepreg (especially in plane shear) was used for the case study. This is because, in such a case, it would be critical for test candidates to get some form of haptic feedback during training

that would give them an idea of how much force/pressure must be applied to the prepreg to obtain the required deformations. Since the current VR training aid does not include any form of haptic feedback, it may not be suitable to train laminators on the layup of more complex shapes. There is certainly room to explore the use of haptic feedback with VR for hand lamination training. However, the author decided not to go down this path for the rest of the duration of this PhD project, since there also exists few major bottlenecks in the concept of such VR training simulations as explained below.

As mentioned in section 4.3.2 of this chapter, all deformations of the prepreg designed in to the VR simulation are pure animations that were predetermined to mimic as closely as possible the actual deformation of the prepreg. This means that the user does not have the opportunity to actively interact with the virtual PJN to try out different layup sequences. The animation approach also means that each step of the VR simulation must be manually coded in to Unity 3D gaming engine. This was found to be a very time consuming and an inefficient process and there is very little room to automate the generation of VR training simulations for a given mould shape following this approach. The solution to these drawbacks would be the design of an interactive PJN that would deform actively based on the user input.

In addition, currently there is no mechanism to decide which layup sequence is ideal for a given mould shape. The sequence of actions used for the VR training simulation presented in this chapter was chosen following video analysis of expert laminators laying up the particular mould used in this case study. Upon the realization of these limitations, the author decided to explore solutions to these limitations instead of further exploring the use of VR technology for hand lamination training. In the remaining chapters of this thesis, an artificially intelligent layup agent is introduced and explored as a solution to these limitations.

CHAPTER 5: An Artificially Intelligent Layup Agent: Preliminary Case Study (Modelling)

5.1 Introduction

In this chapter, the modelling process for a preliminary case study on the use of computational reinforcement learning and convolutional neural networks to create a self-learning artificially intelligent composite layup agent will be presented. The mould shape used for this preliminary case study has been intentionally kept very simple in order to first understand the working mechanics of such a learning algorithm. The mould shape used in this case study contains a flat surface followed by a ramp at 45^{o}, which requires simple physical manipulation to a woven cloth to be laid up properly. The woven reinforcement is modelled as a rigid pin-jointed net. The layup process is modelled as a video game with a digital agent playing it. The goal of the agent is to learn through trial and error, what the best possible sequence of actions (or sequences of actions) required to layup the particular mould it has been presented with. Layup of more complex shapes will be explored in subsequent chapters of this thesis.

5.2 Modelling the Reinforcement Learning System

In this section, detailed explanations on how each of the six main elements of a reinforcement learning system as mentioned in section 2.8 was modelled is provided.

5.2.1 The Closed Environment

The closed environment of the reinforcement learning system was modelled completely in the Unity 3D gaming engine [131]. This particular gaming engine was chosen mainly due to its shallow learning curve and the author's previous experience in its use. There are two main objects in the environment:

(1) The Mould: Dimension of the mould used are shown in Figure 5.1. Note that no units have been used on dimensions. This is because absolute units do not matter for this particular simulation and only the relative units matter.

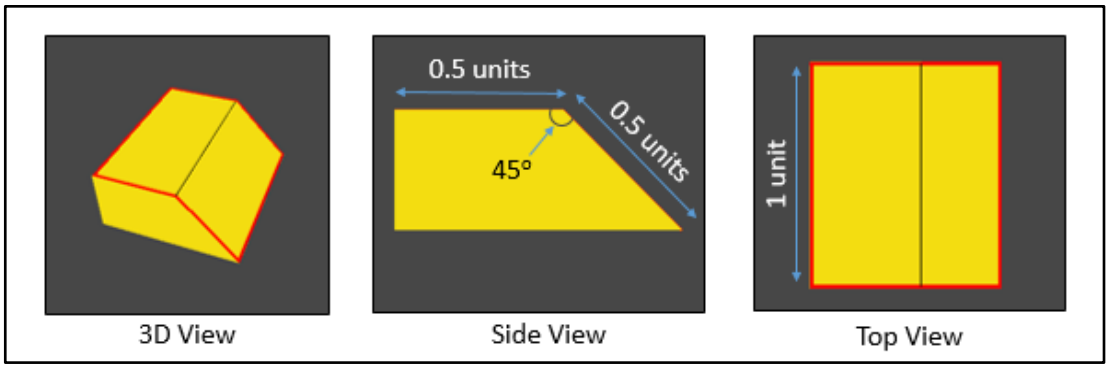


Figure 5.1 - Dimensions of the mould used for the preliminary case study

(2) The Woven Reinforcement: The reinforcement was modelled as a rigid pin-jointed net with a resolution of 3x3, that is nine nodes in total. Each spherical node has a diameter of 0.08 units and each bar connecting two nodes has a length of 0.5 units. Figure 5.2 illustrates these dimensions and other relevant information on the model of the reinforcement.

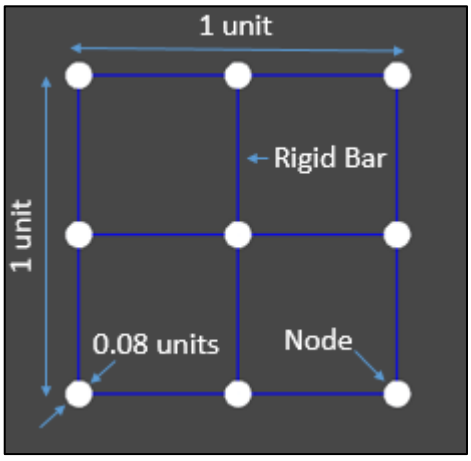


Figure 5.2 - Model and dimensions of the woven reinforcement

The Pin-Jointed Net Model

Deformation of woven clothes occur by rotating warp and weft tows relative to each other [132]. For the most part, woven clothes deform as if the warp and weft tows were

pinned together at the cross-over points and the deformation of the woven reinforcement can be modelled quite closely using a simple pin-jointed net (PJN) [133]. In a pin-jointed net, nodes are connected to each other via inextensible rigid bars with constant length. Other than out of plane bending, the dominant form of deformation of woven reinforcement is trellis type in plane shear. While out of plane bending can be modelled with high accuracy using a PJN model, trellis shear can only be modelled up to a certain limit. This limit is defined by a locking angle and the cloth usually undergoes wrinkling beyond this point. Therefore, wrinkling cannot be modelled using a PJN model [134]. In addition, other micro deformation modes such as fibre slippage present in woven clothes cannot be modelled using a basic PJN model either. However, efforts have been made to incorporate fibre slippage in to PJN models [135]. Nevertheless, for the purpose of this preliminary case study, where the main focus is on modelling the reinforcement learning process, a simple PJN model for the woven reinforcement capable of modelling both out of plane bending and in plane shear is deemed to be sufficient by the author.

In addition to these two main distinct objects, a key component in the Unity 3D gaming engine called ‘Colliders’ are used to detect interactions between these two objects. Colliders were used to detect three interaction modes between the PJN and the mould and to set a boundary to ensure that the environment remains closed. Colliders are represented by green wire boxes in the Unity 3D gaming engine.

Interaction mode 1 between the PJN and the mould: Gain and loss of contact

A mesh collider is attached to the mould. Mesh colliders are generated automatically in the Unity 3D gaming engine and they match exactly to the shape of the CAD model. These colliders can be viewed in Unity 3D by disabling the mesh of CAD models. Figure 5.3 illustrates the auto generated mesh collider attached to the CAD model of the mould.

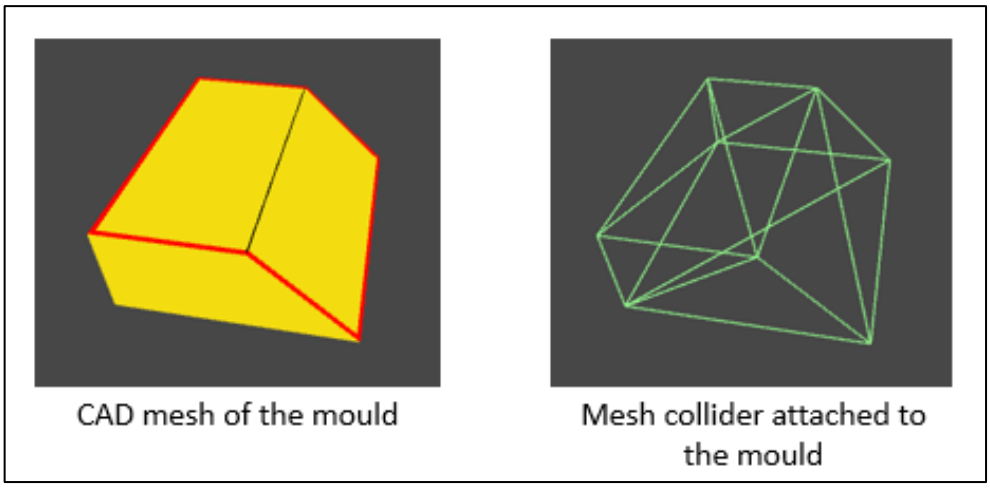


Figure 5.3 - Representation of the mesh collider attached to the mould

Every node of the PJN also contains a collider. For this case, spherical colliders with same dimensions as the nodes were used. This is illustrated in Figure 5.4. It should be noted that no colliders were attached to the rigid bars connecting nodes to each other.

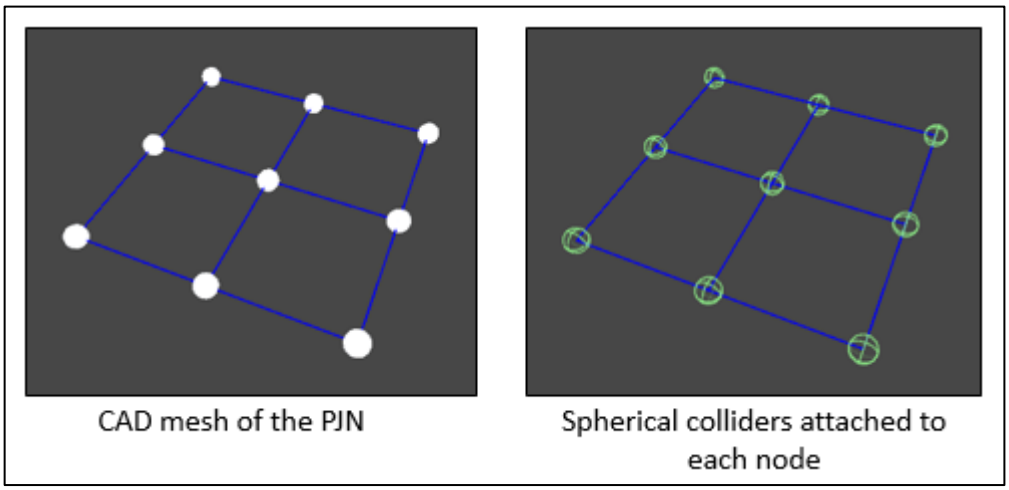


Figure 5.4 - Representation of spherical colliders attached to each node of the pin jointed net

The Unity 3D gaming engine allows detection of contact between nodes and the mould, anytime a spherical collider attached to nodes intersect with the mesh collider attached to the mould. Gain of contact of nodes with the mould is identified visually by changing the colour of the respective nodes from white to black. Therefore, all free nodes are represented in white colour and any node that has gained contact with the mould, or in other words has been stuck on to the mould is represented in black colour. When the

net is hanging freely on top of the mould with no nodes in contact with the mould, all nodes will be white in colour, and if the net was simply moved down until, it gained contact with the flat region of the mould on the top, we can expect few nodes to change colour from white to black. This is illustrated in Figure 5.5. A close up view of this process of collider intersection that causes the colour change of nodes is provided in Figure 5.6. In a subsequent section of this chapter, it will also become clear why such a colour change is critical in the learning process of the agent.

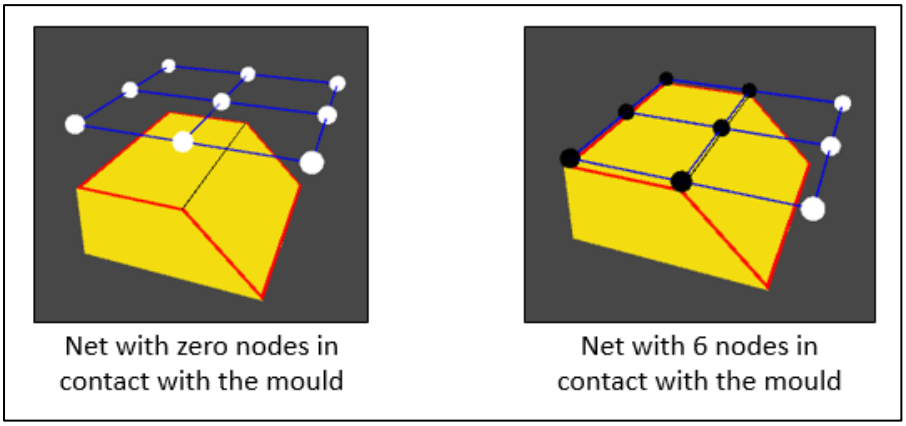


Figure 5.5 - Illustration of net nodes gaining contact with the mould

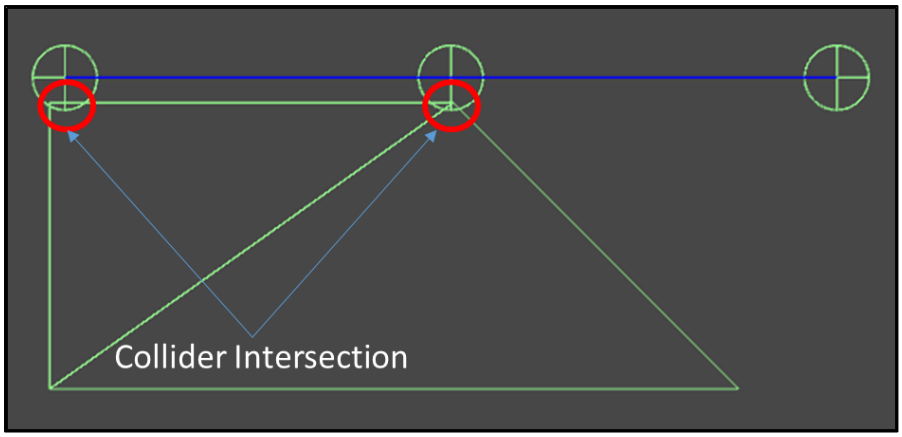


Figure 5.6 - Close up view (side view) of collider intersection

Nodes that are already in contact with the mould can also lose contact with the mould upon application of certain manipulations to the PJN. In such a scenario, the colour of the respective nodes will change from black to white.

Interaction mode 2 between the PJN and the mould: Node submersion in the mould

As shown in Figure 5.6, spherical colliders must go inside of the mesh collider attached to the mould to detect contact between them. There are few cases where nodes can fully submerge in to the mesh collider of the mould. An example of such a case is provided in Figure 5.7, where a diagonal bending action performed on the PJN led to node submersion.

This is clearly unrealistic, since in real life, it is impossible for any part of the woven reinforcement to go pass the physical surface of the mould. Therefore, an extra dummy mesh collider that is a scaled down version of the original mesh collider is placed inside the original mesh collider to detect any node submersions. A node is considered submerged if more than half of its volume is inside the mesh collider attached to the mould. Therefore, the dummy collider is scaled down such that the gap between it and the original mesh collider is equal to half the diameter of a node (0.08/2 = 0.04 units). This is illustrated in Figure 5.8. Detection of node submersion results in the agent being given a negative reward and that particular episode of learning being terminated.

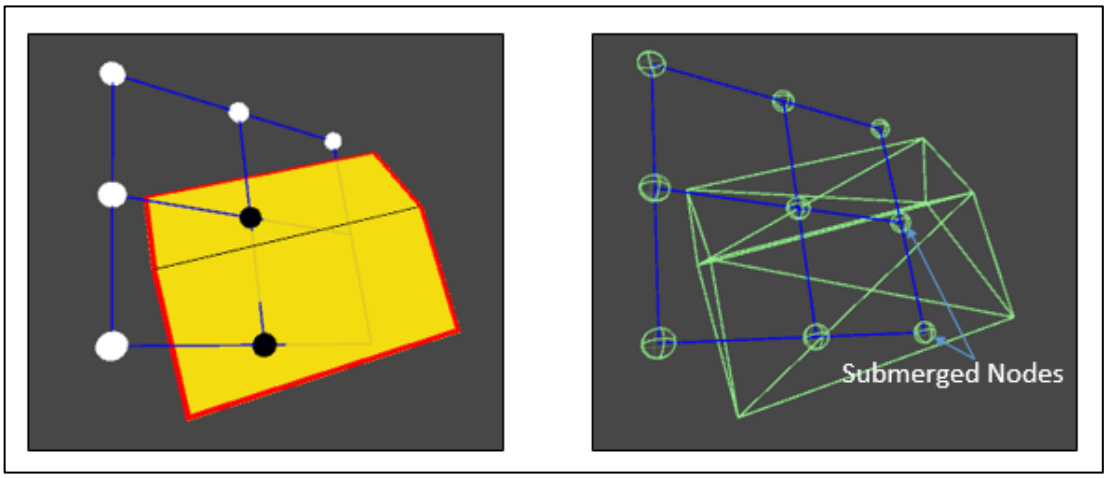


Figure 5.7 - Illustration of node submersion in the mould

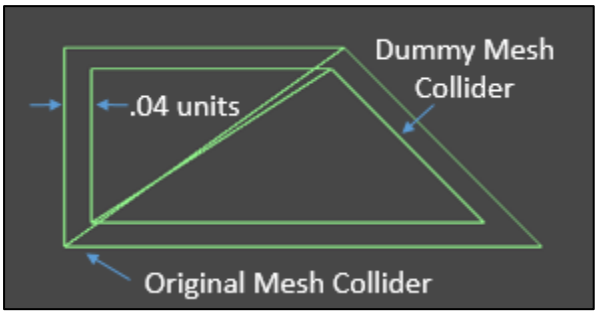


Figure 5.8 - Illustration of the use of a dummy collider to detect node submersion (side view)

Interaction mode 3 between the PJN and the mould: Gain of contact with sides of the mould

The outline marked in red colour in Figure 5.1 represents the boundary within which the net should gain contact with the mould. However, during the agent's learning process, the net also comes in to contact with regions of the mould that are outside of this boundary. The system is designed such that any node making contact with the mesh collider attached to the mould yields a positive reward for the agent. Therefore, additional colliders outside of this boundary had to be used to detect contact between nodes and sides of the mould. Multiple box colliders were placed on all sides of the mould to achieve this. Figure 5.9 illustrates how these box colliders were placed around the mould. Similar to the case in submersion of nodes, a node coming in to contact with any of these additional box colliders would result a negative reward for the agent and is followed by the termination of that particular learning episode. These additional colliders could have been avoided if the agent was restricted in its ability to translate the net away from the mould boundary. This means that the agent only has the ability to move the net Downwards (-Y direction, refer to section 5.2.2 - Translation of the net), and all other translation actions are made unavailable. This type of action restriction may not work for more complex mould shapes that doesn't have an ideal sequence of actions that can be predetermined.

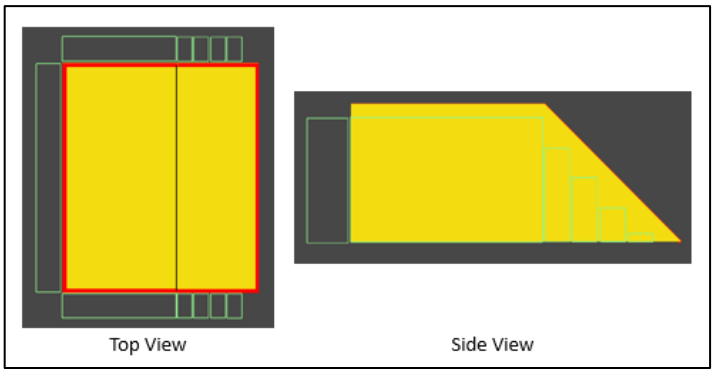


Figure 5.9 - Positioning box colliders to detect contact between nodes and the sides of the mould

Setting the environment boundary with a box collider

In the simulation designed, the agent has the freedom to move the net as a whole in any direction. Therefore, to ensure that the net does not move far away from the mould, a finite boundary had to be set to the environment. A box collider was used to achieve this. The dimensions of this box collider (1.8 x 1.8 x 1.8 in this case), were chosen to ensure that a single translational action away from the mould taken by the agent would result in the net moving out of this collider box, and hence resulting in a negative reward to the agent. Figure 5.10 illustrates this setup. The Unity 3D gaming engine is capable of detecting when a sphere collider that was already inside a box collider leaves it. When this happens in the simulation, the agent is given a negative reward and the current learning episode is terminated and reset to its original state.

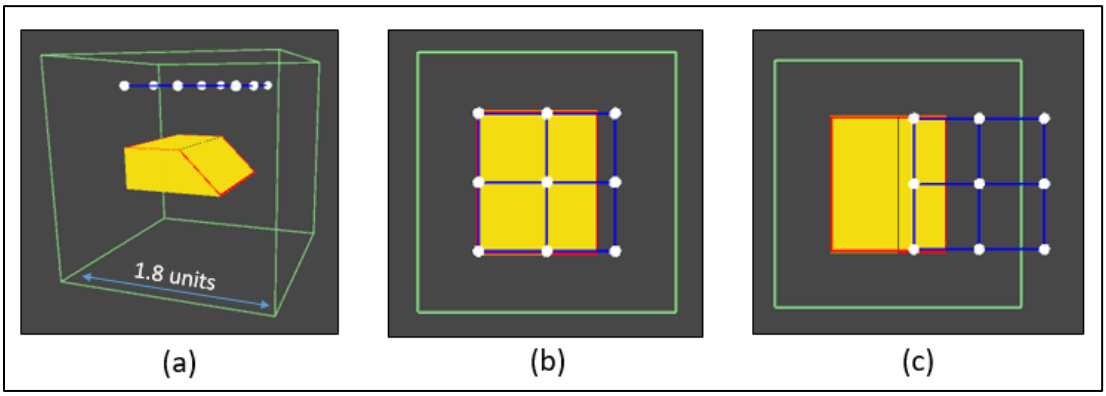


Figure 5.10 - Illustration of the finite boundary set to the simulation environment (a) – 3D view of the finite boundary of the environment, (b) – Top view and (c) – Illustration of an example when the net has moved outside the finite boundary (top view)

5.2.2 Actions the Agent can take in the Reinforcement Learning System

In this particular simulation, all the actions the agent can take are directly linked to the deformations and translations of the net. The total number of actions the agent can take depends on the number of nodes used to model the net. For a net with resolution 3x3, 58 actions are available to the agent. These actions are divided in to four categories: (1) Diagonal out of plane bending of the net, (2) Straight out of plane bending of the net, (3) Translations of the net and (4) In plane shearing of the net. Each of these actions are indexed from 1 to 58 and these indexes are important in understanding the results from the learning process of the agent.

(1) Diagonal out of plane bending of the net (Actions indexed from 1 to 32)

The net can be bent out of plane diagonally in all four diagonal directions. In each diagonal direction, pivots are placed at each point about which the net can be diagonally bent. At this point, the coordinate system of Unity 3D is introduced in Figure 5.11 to make the understanding and notation of diagonal bending of the net more intuitive.

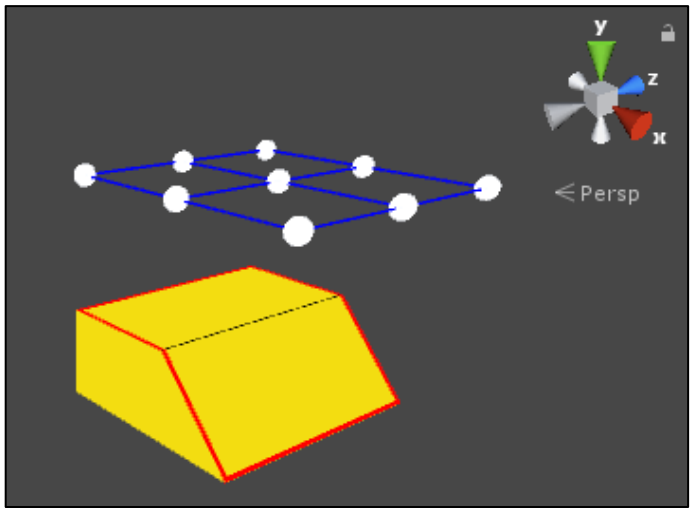


Figure 5.11 - Coordinate system in unity 3D relative to the net and the mould

(Actions 1-4): Downward diagonal out of plane bending about the [-X, -Z] -> [X, Z] direction

4 pivots are used to achieve actions 1-4. A pivot is placed on each node except the last diagonal node and halfway between every two diagonal nodes. This is illustrated in Figure 5.12. The net can be bent about the [-X, -Z] -> [X, Z] direction at each of these pivot points in the downward (-y) direction. The resulting net deformations upon taking each of these actions are illustrated in Figure 5.13. It should be noted that the bending action about the 1st pivot is pure rotation in this case and all the diagonal and straight bending cases explained below. To avoid unnecessary complications in action nomenclature, these pure rotation actions are also labelled as bending actions from this point onwards.

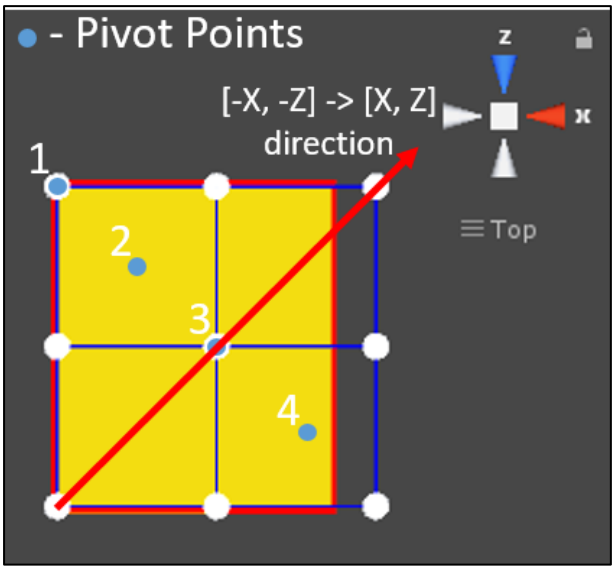


Figure 5.12 - Pivot positions for out of plane downward diagonal bending about the [-X, -Z] > [X, Z] direction

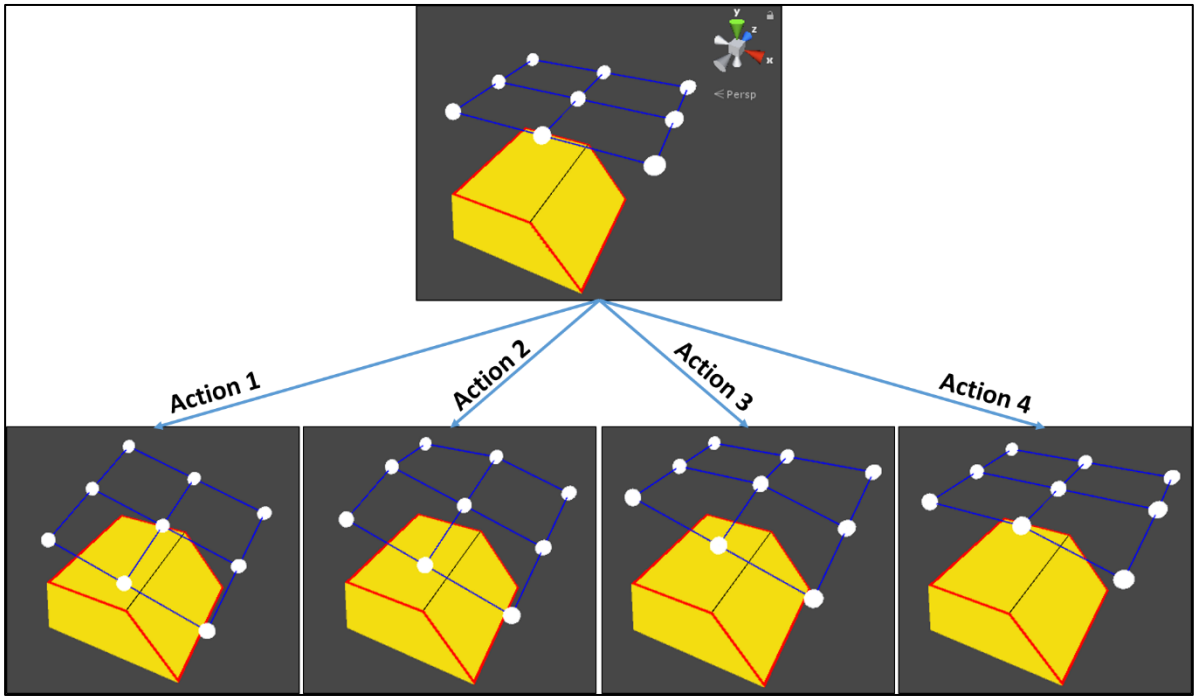


Figure 5.13 – Net deformations from actions 1-4

(Actions 5-8): Upward diagonal out of plane bending about the [-X, -Z] -> [X, Z] direction

Actions 5-8 are identical to actions 1-4, except that the net is rotated about each pivot point in the upward (+y) direction. The resulting net deformations upon taking each of these actions are illustrated in Figure 5.14. The amount by which the net rotates in taking any of these actions is a hyper-parameter that can be defined in the simulation.

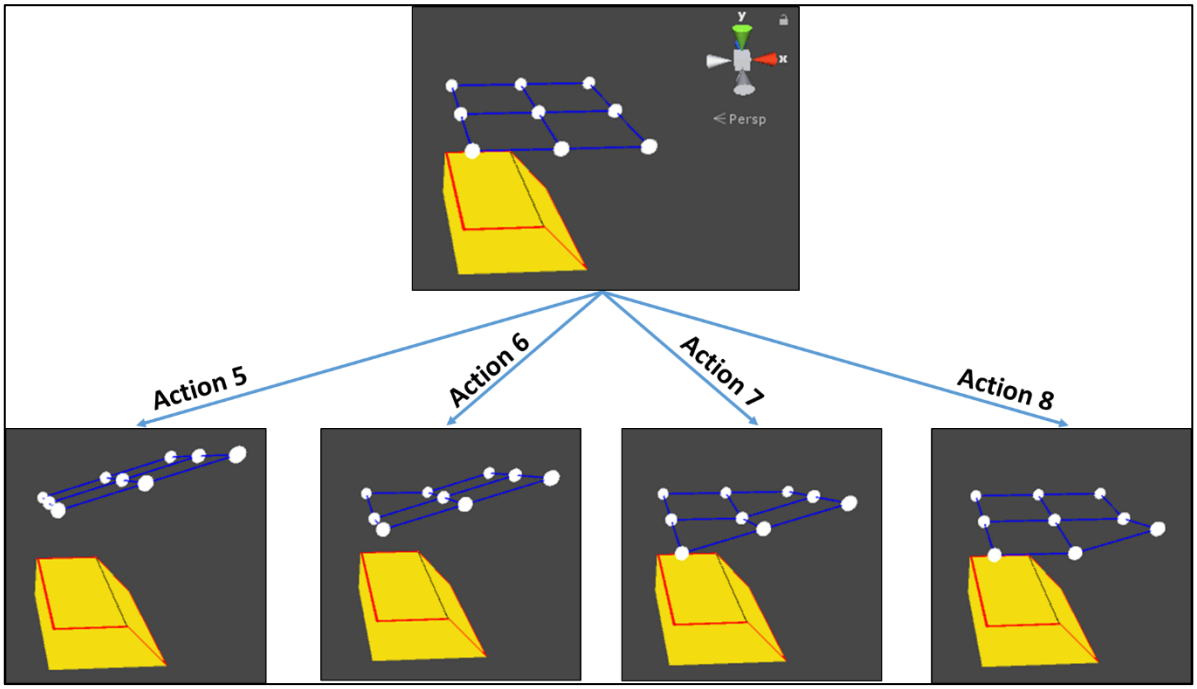


Figure 5.14 - Net deformations from actions 5-8

Actions indexed from 1-8 represent all possible diagonal out of plane bending actions in both up and down directions for one diagonal direction of the net. The net can be bent out of plane in an identical manner in the remaining three diagonal directions. Graphical representations for actions 9-32 are not provided here. Instead, they are defined as:

- Actions 9-12: Downward diagonal out of plane bending about the [X, Z] -> [-X, -Z] direction.
- Actions 13-16: Upward diagonal out of plane bending about the [X, Z] -> [-X, -Z] direction.

- Actions 17-20: Downward diagonal out of plane bending about the [X, -Z] -> [-X, Z] direction.
- Actions 21-24: Upward diagonal out of plane bending about the [X, -Z] -> [-X, Z] direction.
- Actions 25-28: Downward diagonal out of plane bending about the [-X, Z] -> [X, -Z] direction.
- Actions 29-32: Upward diagonal out of plane bending about the [-X, Z] -> [X, -Z] direction.

(2) Straight out of plane bending of the net (Actions indexed from 33 to 48)

The net can be bent out of plane and straight in four directions (+Z, -X, -Z and +X). In each direction, pivots are placed at each point about which the net can be bent.

(Actions 33-34): Downward straight out of plane bending about the +Z direction

2 pivots are used to achieve actions 33-34. A pivot is placed on each mid node, except the last node. This is illustrated in Figure 5.15. The net can be bent about the +Z direction at each of these pivot points in the downward (-y) direction.

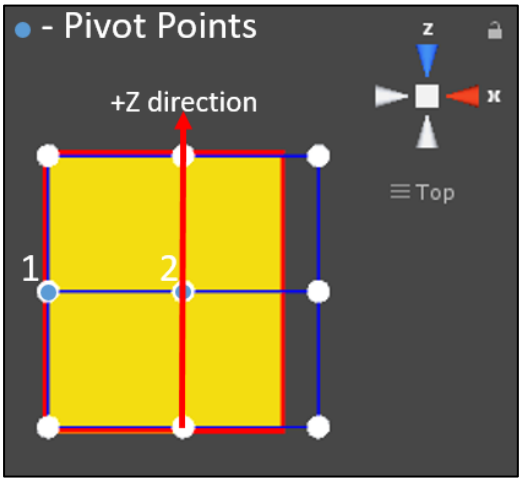


Figure 5.15 - Pivot positions for out of plane downward straight bending about the +Z direction

The resulting net deformations upon taking each of these actions are illustrated in Figure 5.16. The amount by which the net rotates can be defined in the simulation through a hyper-parameter.

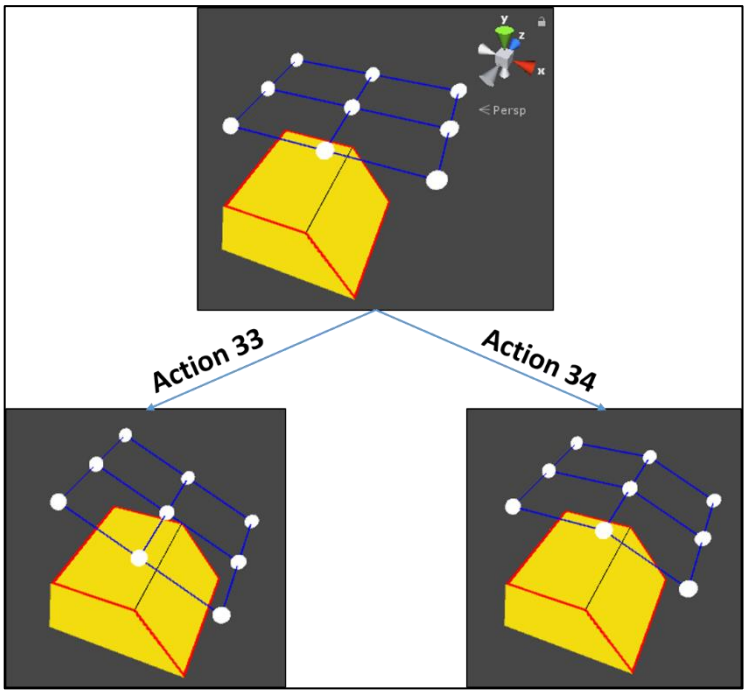


Figure 5.16 – Net deformations from actions 33-34

(Actions 35-36): Upward straight out of plane bending about the +Z direction

Actions 35-36 are identical to actions 33-34, except that the net is rotated about each pivot point in the upward (+y) direction. The resulting net deformations upon taking each of these actions are illustrated in Figure 5.17.

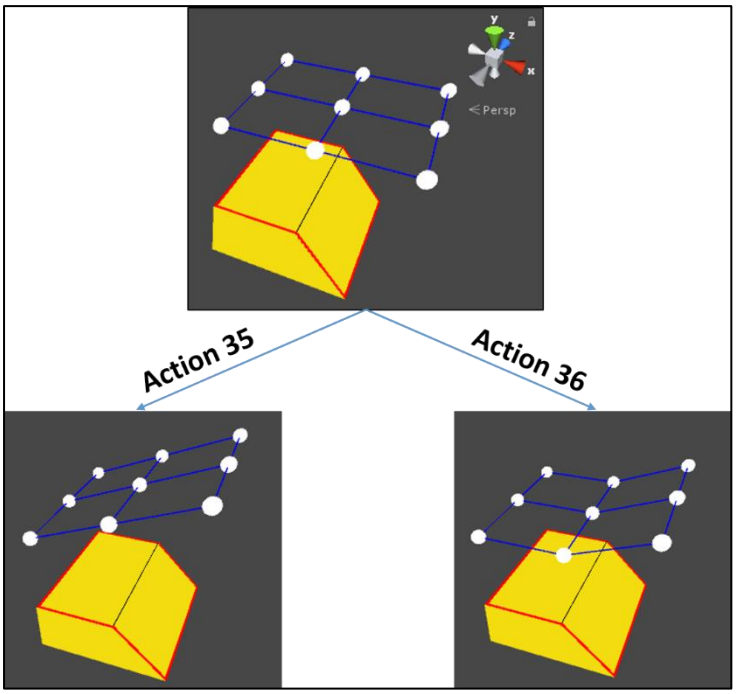


Figure 5.17 - Net deformations from actions 35-36

Actions indexed from 33-36 represent all possible straight out of plane bending actions in both up and down directions for '+Z' direction. The net can be bent out of plane in an identical manner in the remaining 3 directions (-X, -Z and +X). Graphical representations for actions 37-48 are not provided here. Instead, they are defined as:

- Actions 37-38: Downward straight bending about the –X direction.
- Actions 39-40: Upward straight bending about the –X direction.
- Actions 41-42: Downward straight bending about the –Z direction.
- Actions 43-44: Upward straight bending about the –Z direction.
- Actions 45-46: Downward straight bending about the X direction.
- Actions 47-48: Upward straight bending about the X direction.

(3) Translation of the net (Actions indexed from 49 to 54)

The net can be translated in 6 directions: +Y (Up), -Y (Down), +X (Right), -X (Left), +Z (Forward) and –Z (Backwards). These actions are illustrated in Figure 5.18. The amount by which the net gets translated is controllable through a hyper-parameter of the simulation.

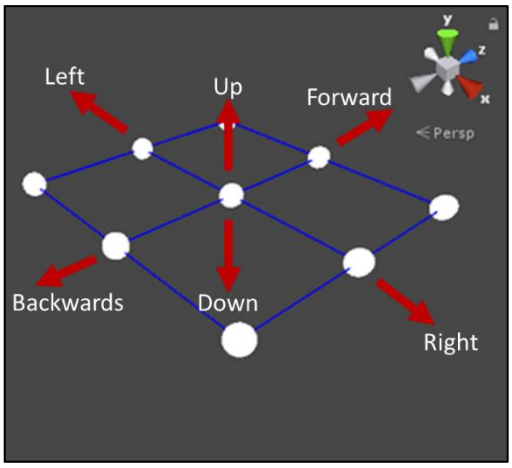


Figure 5.18 - Translation actions on the net (Action 49: Up, Action 50: Down, Action 51: Right, Action 52: Left, Action 53: Forward and Action 54: Backwards)

(4) In plane shearing of the net (Actions indexed from 55 to 58)

The net can be sheared in plane in all four diagonal directions. The amount of shear applied in each action is controlled by a hyper-parameter that can be set in the simulation. Prepreg material shear in a certain progressive manner. I.e. when modelling the woven prepreg with a PJN under shear, the net should not shear globally. The shear deformation is transferred progressively from the point of application of shear [53]. Both these forms of shear deformations are illustrated in Figure 5.19 and Figure 5.20.

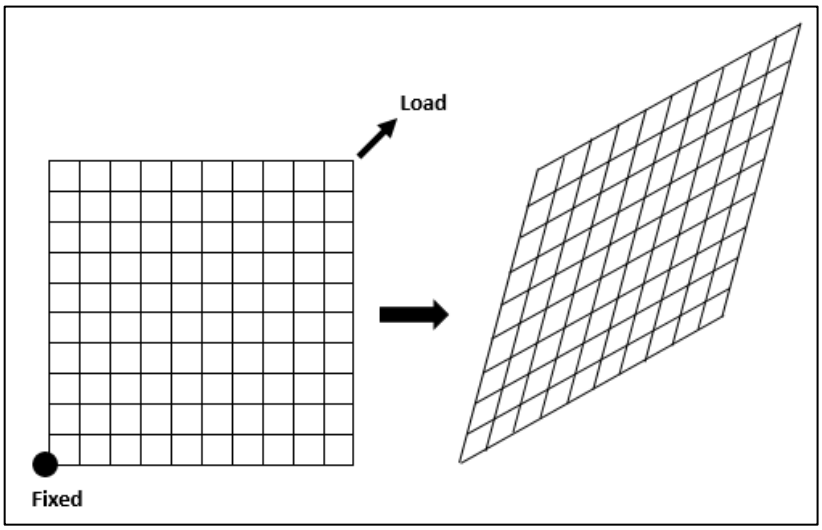


Figure 5.19 - Global shear deformation of a pin jointed net

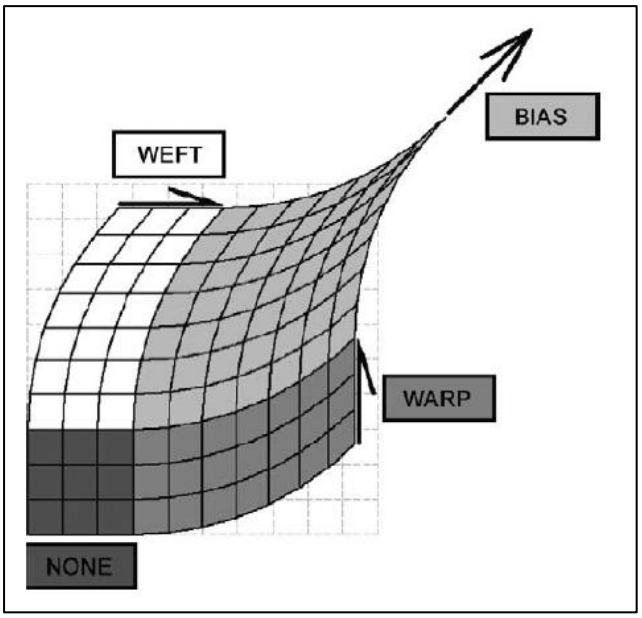


Figure 5.20 - Progressive shear deformation of a pin jointed net

The Locking Angle of a Unit Cell in the PJN

It was previously mentioned that the PJN could not further deform under shear once a certain locking angle is reached. This concept is applicable to each unit cell of the PJN. In this section, a single unit cell of the PJN as illustrated in Figure 5.21 will be used to explain the concept of the locking angle. As the load applied on the unit cell is increased progressively, the angle θ reduces. Once it reaches a critical locking angle θ_L, the unit cell can no longer be sheared from this point onwards. The cell is supposed to wrinkle beyond this point. But, as mentioned previously, wrinkling is not modelled and instead the agent is given a negative reward if it tries to shear any unit cell of the PJN beyond its locking angle. The locking angle for a particular woven reinforcement type has been shown to be a function of initial tow width and pitch [136].

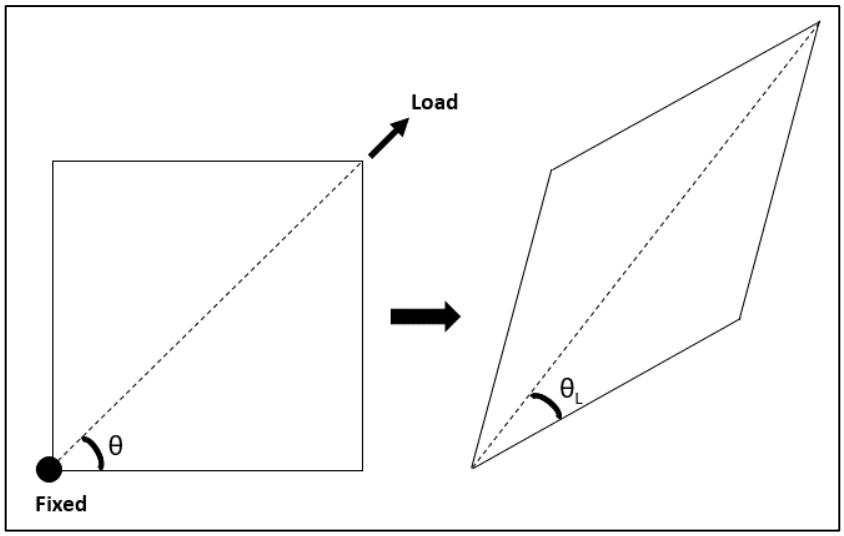


Figure 5.21 - Locking angle of a single unit cell in a pin jointed net

Modelling the Progressive Shear Deformation of the PJN

In order to explain the modelling process of the progressive shear deformation of a PJN, a net with 3x3 resolution, as illustrated in Figure 5.22 will be used. In this illustration, node 1 is fixed and a load is applied diagonally in the 1->9 direction at node 9. Starting angle of θ is 45°.

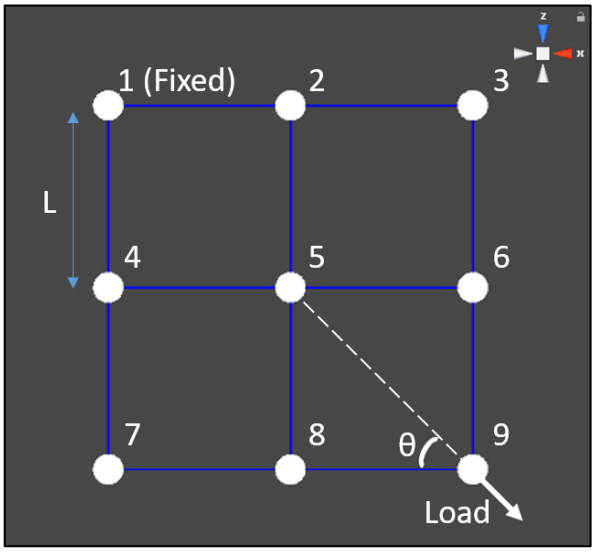


Figure 5.22 – Pin jointed net with 3x3 resolution to demonstrate the progressive shear model

The shear deformation of the net in a particular diagonal direction is defined using the total diagonal length in that direction. In the simulation, this diagonal length is increased incrementally to shear the net. Consider shearing the net shown in Figure 5.22 in the 1->9 direction. Let the initial diagonal length between nodes 1 and 9 be D. Now, increment this length by a small amount 'd'. The steps in calculating the resulting node positions of the net are given below. It should be noted that the equations used in the steps below are only valid for θ in the range of 0^o to $+90^o$, and this is the physically available range for the material due to its woven nature.

Step 1: Calculate maximum allowed distance between two consecutive diagonal nodes (Diag_max) based on the locking angle:

- $Diag\_max = 2*L (\cos \theta_L)$

Step 2: Calculate the total maximum diagonal displacement allowed (Diag_max_total):

- Diag_max_total = Diag_max * Number of diagonal cells = Diag_max * 2

Step 3: Check [accumulation of d + D] against Diag_max_total:

- If [accumulation of d + D] is greater than Diag_max_total, the agent is awarded a negative reward and the net retains its current deformation pattern, since this leads to wrinkling.
- If [accumulation of d + D] is less than Diag_max_total, move to Step 4

Step 4: Determine new positions of the diagonal nodes in the order 9, 5, 1

- $[X\_New_9, Z\_New_9] = [X\_Current_9 + d (\cos 45^o), Z\_Current_9 – d (\sin 45^o)]$
- Calculate diagonal distance between new position of node 9 and current position of node 5, $D_{9\_New, 5\_Current}$.
- If $D_{9\_New, 5\_Current} > Diag\_max$,
  $[X\_New_5, Z\_New_5] = [X\_New_9 - Diag\_max * (\cos 45^o), Z\_New_9 + Diag\_max * (\sin 45^o)]$.
  Else,
  $[X\_New_5, Z\_New_5] = [X\_Current_5, Z\_Current_5]$

- In this case, node 1 is defined to be fixed. However, if a net with higher resolution was used, the same procedure as explained will be used to calculate the new positions of subsequent diagonal nodes.

Step 5: Calculate positions of nodes below the diagonal in the order – (Nodes 8 and 4 first followed by node 7)

- Consider node 8 first:
  Calculate new diagonal distance between nodes 5 and 9 ($D_{5\_New,\ 9\_New}$)
  Calculate the angle enclosed by nodes 5, 9 and 8 (θ), $\theta = \cos^{-1} (D_{5\_New,\ 9\_New} / [2L])$
  $[X\_New_8, Z\_New_8] = [X\_New_9 – L * \cos (45-\theta), Z\_New_9 + L * \sin (45-\theta)]$
- New position of node 4 can be calculated in a similar manner to node 8 using new positions of nodes 5 and 1 calculated in step 4.
- New position of node 7 can be calculated using new positions of nodes 8 and 4 in a similar manner to how new positions of nodes 8 and 4 were calculated.

Step 6: Calculate positions of nodes above the diagonal in the order – (Nodes 6 and 2 first followed by node 3)

- New positions of nodes above the diagonal are calculated in an ideal manner to how new positions of nodes below the diagonal are calculated.

Once steps 1 to 6 are completed, the incremental shear deformation of the net in one particular direction can be defined. Same steps are used to calculate the shear deformation in all other 3 diagonal directions. Graphical representation of these shear deformations are illustrated in Figure 5.23 and Figure 5.24.

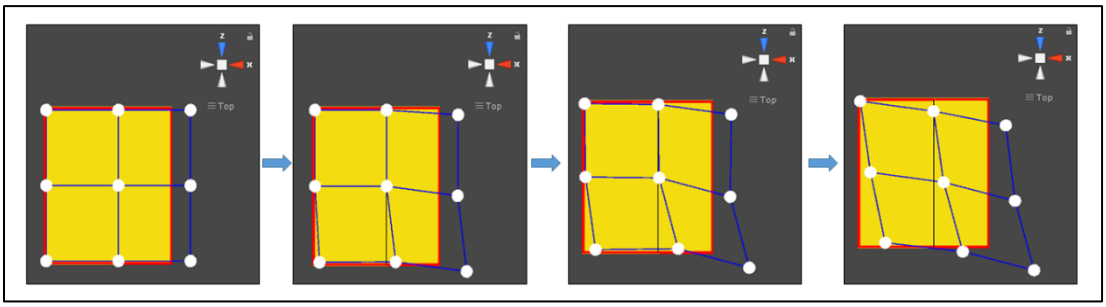


Figure 5.23 - Illustration of progression of shear in the [-X, Z] -> [X, -Z] direction

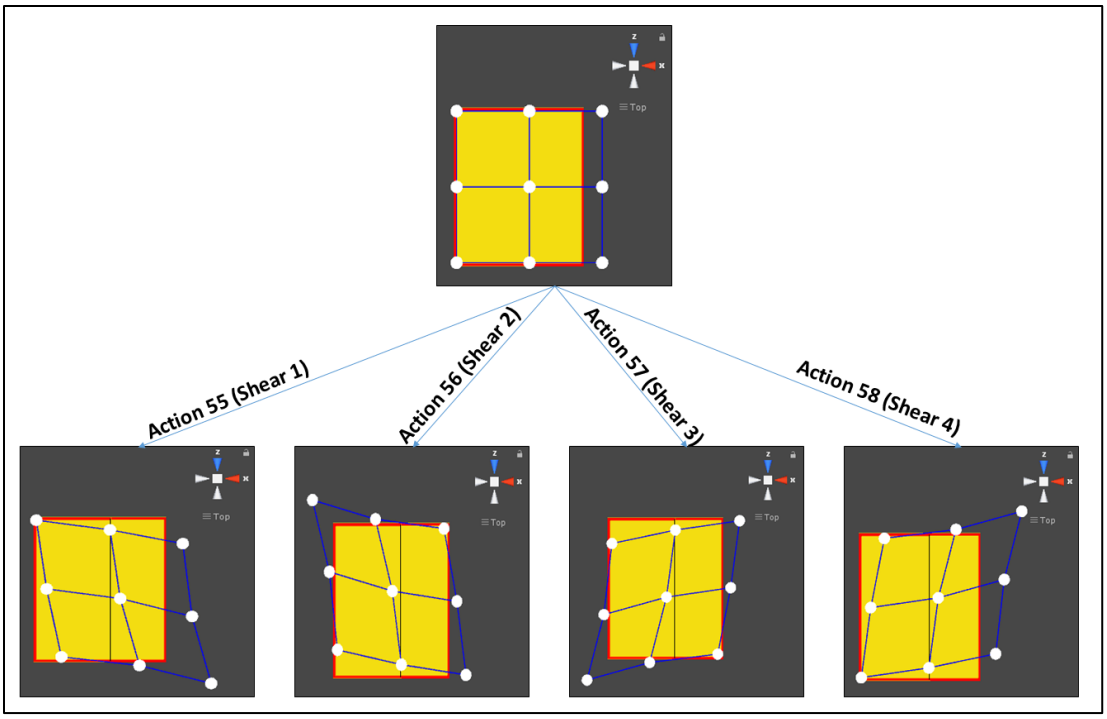


Figure 5.24 - Net deformation patterns from actions 55-58

5.2.3 The Reward System

The simulation is set up such that the agent receives positive rewards for desirable actions it takes, negative rewards for undesirable actions and finally zero rewards for neutral actions. All these three categories are described below:

Positive Rewards

There is only one way, the agent can receive a positive reward in this simulation. That is by gain of contact of nodes with the mould. The total cumulative positive reward per episode is set to +1. That means for each node that gains contact with the mould, the

agent receives a positive reward of 1/9 ≈ 0.111. Figure 5.25 shows an example of an action that result in a positive reward for the agent. In this example, the net is initially lying above the mould and the agent has decided to translate the net downwards, and as a result, six nodes have gained contact with the mould, resulting in a positive reward of 6/9 ≈ 0.666.

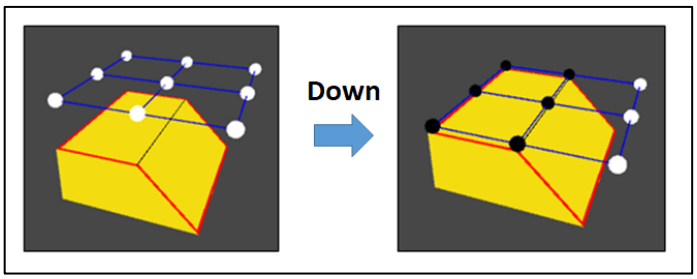


Figure 5.25 - Example of an action that result in a positive reward

Negative Rewards

The agent can receive a negative reward in multiple ways:

(1) Loss of contact of nodes with the mould: Any action that results in nodes that are already in contact with the mould to lose their contact with the mould results in the agent receiving a negative reward. An example of such a situation is shown in Figure 5.26. In this case, the net was already placed on the mould and the agent has decided to take an in plane shear action which has resulted in one of the nodes becoming free. This action will therefore result in a negative reward of -1/9 ≈ - 0.111.

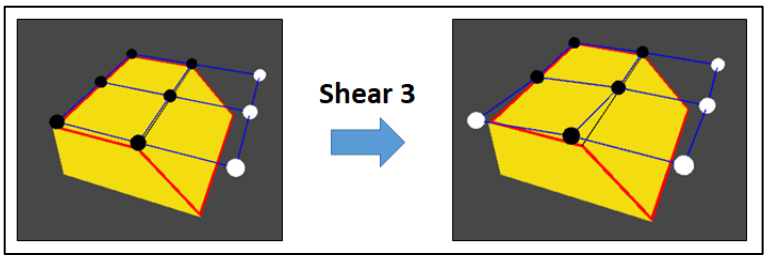


Figure 5.26 - Example of an action resulting in a negative reward because of loss of contact of a node with the mould

(2) Net leaving the closed environment: As mentioned in section 5.2.1 of this chapter, a box collider is used to set a finite volume for the reinforcement-learning environment. If any action taken by the agent results in one or more nodes to leave this finite volume, a negative reward proportional to the number of nodes that left this finite volume will be awarded to the agent. An example of such a case is shown in Figure 5.27. In this case, the agent has moved the net in the +X (right) direction, which has caused 3 nodes to leave the environment boundary shown by the green wireframe. This particular action will result in a negative reward of -3/9 ≈ -0.333 being awarded to the agent. This will also result in the termination of the current learning episode.

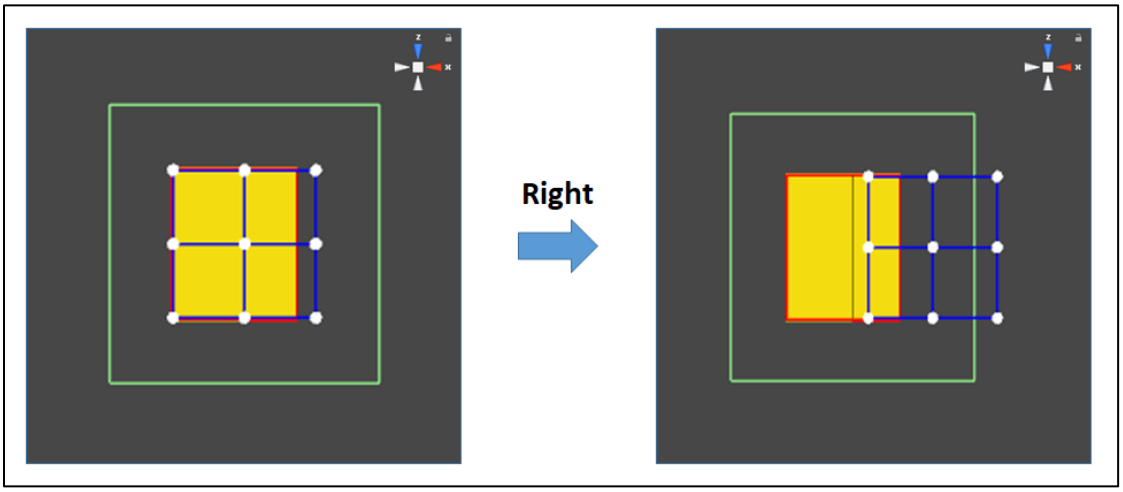


Figure 5.27 - Example of an action resulting in a negative reward because of one or more nodes leaving the finite volume of the reinforcement learning environment

(3) Attempts to shear the net beyond the locking angle: If the agent tries to shear the net in particular direction, once the net has reached its locking angle in that particular diagonal direction, the agent will be awarded a negative reward of -1.

(4) Node submersion: If more than one or more nodes submerges in the mould because of an action taken by the agent as explained in section 5.2.1 of this chapter, it will be awarded a negative reward proportional to the number of nodes that submerges in the mould. In the example shown in Figure 5.7, two nodes have submerged in the mould, and this will result in a negative reward of -2/9 ≈ -0.222.

(5) Translation of the fixed net: The simulation is set up in such a way that if the agent attempts to translate the net with one or more fixed nodes in any of the six directions, a reward of -1 is awarded. Figure 5.28 illustrates an example of such a situation. In this case, the net has already been placed on the mould with six nodes stuck down to the flat region of the mould and the agent attempts to translate the net in the downward direction. Such an action is physically impossible. Therefore, in order to discourage the agent in taking such actions in the future, a negative reward is awarded. A negative reward would be awarded for all other 5 translational actions as well even though those actions are achievable in reality but will lead to loss of contact between nodes and the mould. However, these actions are restricted in the current simulation to speed up the learning process of the agent.

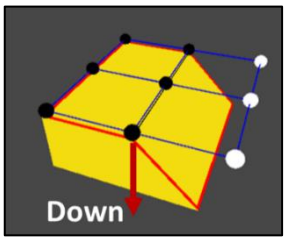


Figure 5.28 – Example of a null action resulting in a negative reward to the agent

(6) Wrinkling due to bending coupling: Another cause of wrinkling is due to bending coupling. There are three forms of bending coupling that cause wrinkling and they will be explained using a PJN with resolution 4x4 as illustrated in Figure 5.29.

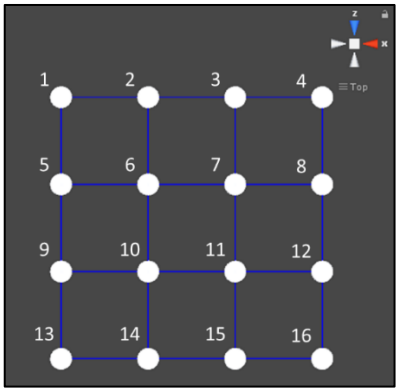


Figure 5.29 – Pin jointed net with resolution 4x4 used to understand bending coupling

(i) Straight – straight bending coupling: Firstly, perform a straight bending of the net at the pivot placed halfway between nodes 7 and 11 about the +Z direction in either the downward or upward direction. Once this bending action has been performed,

- The net can still be bent straight in both upward and downward directions about the –Z direction at any of the pivots placed halfway between nodes [8, 12], [7, 11], [6, 10] and [5, 9] without wrinkling.
- The net can be bent straight in both upward and downward directions at pivots placed halfway between nodes [2,3] and [14, 15] about the directions +X and –X respectively without wrinkling.
- The net can no longer be bent in either the upward or downward directions at pivots placed halfway between nodes [6, 7] and [10, 11] about either directions +X or –X without wrinkling. Therefore, these particular actions are restricted in the simulation and any attempt made by the agent to perform these actions will result in a reward of -1.

(ii) Diagonal – diagonal bending coupling: Firstly, perform a diagonal bending of the net at the pivot placed halfway between nodes 15 and 12 about the [-X, -Z] -> [X, Z] direction in either the downward or upward direction. Once this bending action has been performed,

- The net can still be bent diagonally in either upward or downward direction at the pivots placed on nodes 16, 11, and 6 and pivots placed halfway between nodes [12, 15], [7, 10], and [2, 5] about the [X, Z] -> [-X, -Z] direction without wrinkling.
- The net can still be bent diagonally at the pivots placed on nodes 13, 10, and 7 and halfway between nodes [13, 10] and [4, 10] in both upward and downward directions about the direction [X, -Z] -> [-X, Z] without wrinkling.
- The net can still be bent diagonally at the pivots placed on nodes 4, 7, and 10 and halfway between nodes [4, 7] and [10, 13] in both upward and downward directions about the direction [-X, Z] -> [X, -Z] without wrinkling.
- The net can no longer be bent diagonal in either the upward or downward directions at the pivot placed halfway between nodes 7 and 10 about both directions [X, -Z] -

> [-X, Z] and [-X, Z] -> [X, -Z] without wrinkling. Therefore, these particular actions are restricted in the simulation and any attempt made by the agent to perform these actions will result in a reward of -1.

(iii) Straight – diagonal bending coupling: Firstly, perform a straight bending of the net at the pivot placed halfway between nodes 6 and 10 about the +Z direction in either the downward or upward direction. Once this bending action has been performed,

- The net can be bent diagonally in both upward and downward directions at the pivots placed on nodes 11 and 1 and halfway between nodes [16, 11] and [6, 1] about the direction [-X, -Z] -> [X, Z] without wrinkling, but not at pivots placed at node 6 and halfway between nodes 6 and 11. A similar pattern emerges in trying to bend the net diagonally about the direction [X, -Z] -> [-X, Z].
- The net can be bent diagonally in both upward and downward directions at the pivots placed on nodes 16 and 11 and halfway between nodes [16, 11] and [6, 1] about the direction [X, Z] -> [-X, -Z] without wrinkling, but not at pivots placed at node 6 and halfway between nodes 11 and 6. A similar pattern emerges in trying to bend the net diagonally about the direction [-X, Z] -> [X, -Z].

(iv) Diagonal – straight bending coupling: Firstly, perform a diagonal bending of the net at the pivot placed at node 11 about the [-X, -Z] -> [X, Z] direction in either the downward or upward direction. Once this bending action has been performed,

- The net can be bent straight in both upward and downward directions at pivots placed halfway between nodes [6, 10] and [5, 9] about the +Z direction without wrinkling, but not at the pivot placed halfway between nodes 7 and 11. A similar pattern emerges in trying to bend the net straight about the +X direction.
- The net can be bent straight in both upward and downward directions at pivots placed halfway between nodes [8, 12] and [6, 10] about the -Z direction without wrinkling, but not at the pivot placed halfway between nodes 7 and 11. A similar pattern emerges in trying to bend net straight about the -X direction.

All four coupling modes are programmed in to the simulation and the agent is awarded negative rewards whenever it attempts to perform an action that would lead to wrinkling.

(7) Nodes coming in to contact with box colliders placed on the sides of the mould: Explained in interaction mode 3 in section 5.2.1.

(8) Nodes coming in to contact with other nodes. The agent is awarded a reward of -1 each time one nodes come in to contact with another node. The learning episode is also terminated in such a case.

Zero Rewards

Any action performed by the agent that does not result in either a positive reward or a negative reward is given a zero reward. An example of such a situation is illustrated in Figure 5.30. In this case, Action 34, which is an out of plane straight bending action is performed when the net has been placed on top of the mould. Because of this action, no nodes gained contact or lost contact, and no wrinkling, submersion or bending coupling occurred. Therefore, the resulting reward is zero.

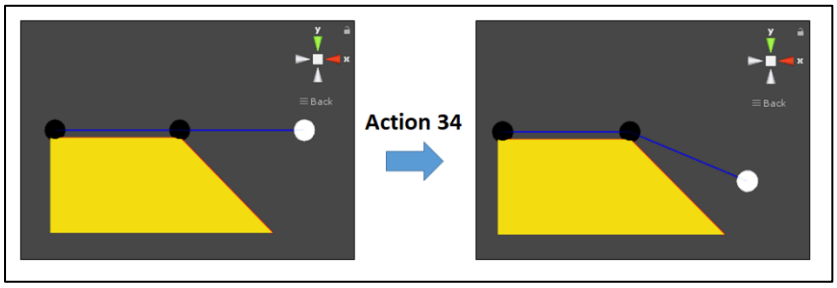


Figure 5.30 - An example of an action taken that results in a zero reward

5.2.4 Goals of the Reinforcement Learning System

In this particular reinforcement learning system, the goal of the agent is to maximize the number of nodes that come in to contact with the mould in the least number of actions as possible.

5.2.5 State Definition

This particular reinforcement learning system uses an image-based state definition. To be more precise, an instantaneous state of the system is defined as a combination of three distinct views (Side, Top and Front) of the environment fine-tuned to capture as much information about the system as possible. In order to achieve this, three cameras are placed in the Unity 3D gaming environment to capture states as illustrated in Figure 5.31. In this figure, two environment states are used to illustrate state representations using three views. In the free net state, the net is hanging with all nodes free above the mould and in the net on mould state, six nodes have come in to contact with the mould.

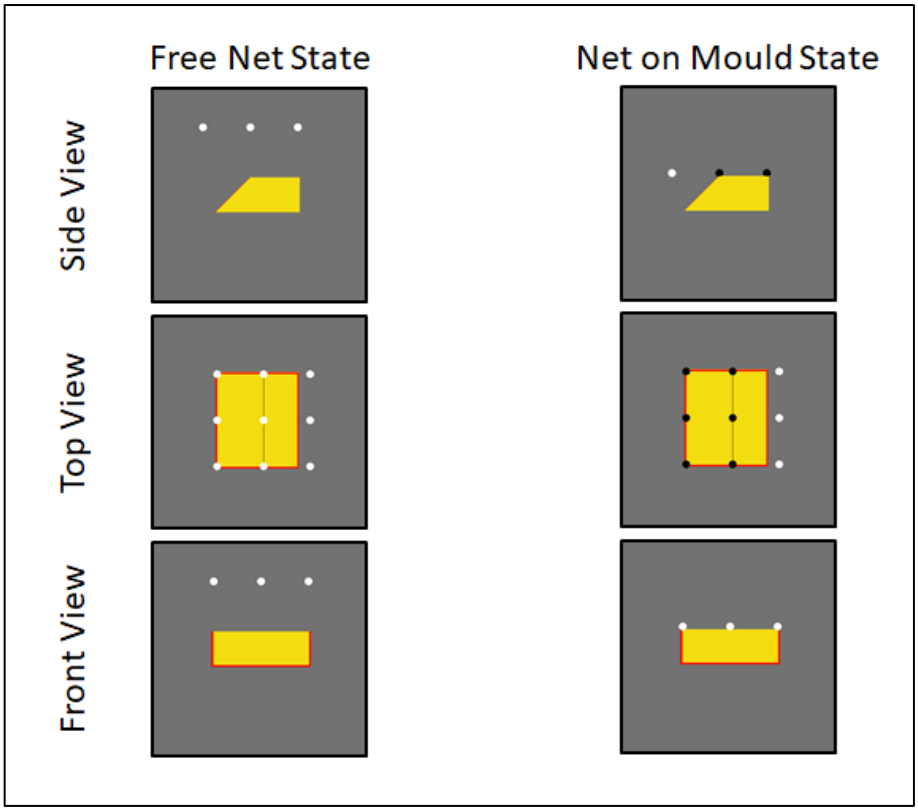


Figure 5.31 - State representations for free net and net on mould states using side, top and front views

5.2.6 The Agent

At each discrete time step, the objective of the agent is to analyse (observe) the current state of the environment and make a smart action that will maximize the future cumulative reward as illustrated in Figure 5.32.

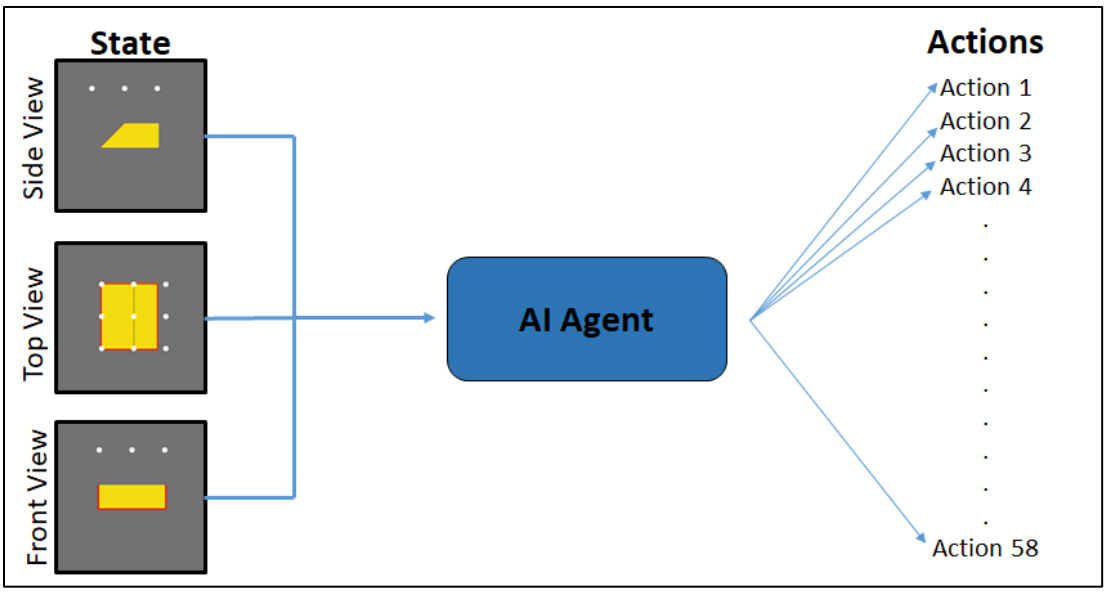


Figure 5.32 - An artificially intelligent agent observing the current state and trying to make a smart action

This situation of the agent is very similar to the situation of a real-world hand laminator. The laminator must analyse the current state of the prepreg, mould, and their combined state and make a smart action that the laminator thinks will result in the best form of final layup. But, for a laminator to be able to make such smart decisions instantly, the laminator had to have gained years' worth of experience and training. The agent is in a similar situation: it must be trained before it is capable of making smart actions. This training process in terms of an algorithm is explained in detail below:

Step 1: Create a model of the environment (Explained in sections 5.2.1, 5.2.1 and 5.2.3 of this chapter)

Step 2: Define the initial state of the environment. The initial state of the environment is the free net state shown in Figure 5.31, with the distance between the net and the flat surface of the mould being 4.5 units. This is illustrated in Figure 5.33.

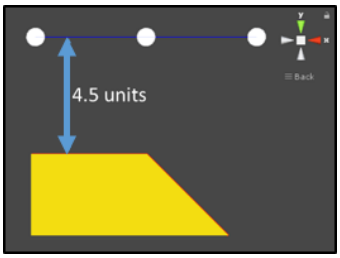


Figure 5.33 - Initial state of the environment

Step 3: Set initial hyper-parameters of the system. These parameters are explained in Table 5.1. Some of these descriptions may not be fully clear at this point. However, they will be explained in more detail further down either in this section or in the next chapter where appropriate.

Table 5.1 - Initial hyper-parameters and their descriptions of the reinforcement learning system

Hyper-Parameter Name	Value	Description
Locking Angle	30°	Locking angle θ_L as illustrated in Figure 5.21.
Rotational Constant	22.5°	The amount by which the net will be rotated at any pivot point in both diagonal and straight bending. Further explained in section 6.4.6.
Camera Frame Size	84	Number of pixels in each side of each of the 3 views describing a state. Further explained in section 6.4.4.
Maximum E Greedy Value	1	Maximum probability with which the agent will take actions in a greedy manner.
Number of Mini Batch Gradient Descents to Maximum E Greedy	75	Number of times the weights of the convolutional neural network will be adjusted (or the number of backpropagation steps) before acting in a fully greedy manner. This probability will be adjusted

Hyper-Parameter Name	Value	Description
		from 0 to 1 in a linear fashion in 75 steps. Further explained in section 6.4.7.
Mini Batch Size	300	Number of state, action, reward, and next state data sets that will be used to perform one backpropagation step.
Number of Minimum Episodes per Mini Batch	30	The minimum number of episodes that will be included in each mini batch used for one backpropagation step.
Maximum Number of Actions per Episode	10	Number of actions the agent is allowed to perform in each episode before the environment is reset to its initial state.
Shear Deformation Step	0.75	The value by which the total diagonal length of the net in a particular diagonal direction is incremented at each shear action. Further explained in section 6.4.6.
Translational Step	4.5	Distance by which the net is translated upon taking any of the 6 translational actions. Further explained in section 6.4.6.
Memory Capacity	3	The number of action sequences the agent can store in its memory to recall in the future. Further explained in section 5.3.3 and section 6.4.8.
Discount Rate (ϒ)	0.99	Explained in section 2.9.4.

(Table 5.1 continued)

Hyper-Parameter Name	Value	Description
Learning Rate (α)	0.01	Explained in section 5.2.6 and 6.4.1.
Weight Swap Steps (C)	300	Explained in section 2.10.3 and 6.4.5.

(Table 5.1 continued)

Step 4: Set up the architecture of the convolutional neural network (CNN) that will act as the main component of the digital brain of the AI agent. The input to the CNN are 3 images corresponding to a state, each with dimensions 84x84. Each of these images has 3 colour channels (red, green and blue). Therefore, the final volumetric dimension of the input is 84x84x9. The output from the CNN approximates the values of each of the 58 possible actions at any given state. Therefore, the output layer of the CNN is a fully connected layer with 58 units. The activation function used is a hyperbolic tangent function (Tanh function). This architecture was adapted from [116]. The final architecture is described in Table 5.2.

Table 5.2 - CNN architecture used for the AI agent

Layer	Name	Input	Function	Output
1	Input Layer	-	-	84 x 84 x 9
2	Convolutional Layer	84 x 84 x 9	Convolution using 256 filters, each with dimensions 8 x 8 x 9. A stride of 4 is used with no zero padding. This is followed by an activation using the Tanh function.	20 x 20 x 256

Layer	Name	Input	Function	Output
3	Convolutional Layer	20 x 20 x 256	Convolution using 256 filters, each with dimensions 4 x 4 x 256. A stride of 2 is used with no zero padding. This is followed by an activation using the Tanh function.	9 x 9 x 256
4	Convolutional Layer	9 x 9 x 256	Convolution using 256 filters, each with dimensions 3 x 3 x 256. A stride of 1 is used with no zero padding. This is followed by an activation using the Tanh function.	7 x 7 x 256
5	Fully Connected Layer	1 x 12544	Multiplication with a matrix with synaptic weights of dimensions 12544 x 2048. This is followed by an activation using the Tanh function.	1 x 2048
6	Fully Connected Layer	1 x 2048	Multiplication with a matrix with synaptic weights of dimensions 2048 x 58. This is followed by an activation using the Tanh function.	1 x 58

(Table 5.2 continued)

Step 5: Initiate synaptic weights of the CNN. Note that at the beginning of the simulation, the value of each single synaptic weight is unknown. Therefore, these values must be set to some initial value prior to training the CNN. It has been established that arbitrary initialization of weights will slow down the training process and sometimes even lead to failure of convergence of the CNN to an accurate state [137]. If the input data were properly normalized, it is reasonable to assume that half of the weights will be positive and the other half negative [138]. Therefore, a reasonable sounding approach would be to set all weights to zero at the beginning. However, this is a mistake, since if every neuron in the network outputs the same value, then they will also have the same error gradient during backpropagation and as a result will undergo the same parameter updates. In other words, there is no asymmetry between the synaptic weight values if they are all initialized at zero.

In order to break symmetry, it is common to initialize the weights of the neurons to random values. Since the final outputs of the CNN are within the range -1 to +1, it is good practice to keep the magnitudes of these weights close to zero. A typical implementation of one matrix of weights (w) might look like,

$$w = 0.01 * \text{RANDN}(\text{Depth}, \text{Height}). \qquad \textbf{Equation 5.1}$$

Where, RANDN samples from a zero mean, unit standard deviation Gaussian distribution. It is also common to draw small numbers from a uniform distribution, but there is very little difference on the final performance of the CNN [138]. It should also be note that making weights smaller does not always improve the performance of the CNN. If the weights are initialized too small, the gradients flowing through the CNN during backpropagation will also be very small and will become a concern especially for deeper networks.

There is a problem in the above approach such that the variance of outputs from a randomly initialized neuron grows with the number of inputs. This problem can be solved by scaling the weight vector by the square root of the respective number of input

units. This also normalizes the variance of each neuron's output to 1. With this understanding, Equation 5.1 can be modified as,

$$w = RANDN (Depth, Height) / SQRT (n). \quad \textbf{Equation 5.2}$$

The derivation of Equation 5.2 can be found in [138]. Another variation to this is proposed in [139], in the form of:

$$w = RANDN (Depth, Height) / (2 / (n_{in} + n_{out})). \quad \textbf{Equation 5.3}$$

Where, n_{in} and n_{out} are the number of neurons in the previous and next layers. Both these methods yield similar results. However, these initialization methods are not so effective for ReLU neurons. A recent paper published on this topic, [140] recommends using the following initialization for neural networks with ReLU units,

$$w = RANDN (Depth, Height) * SQRT (2 / n). \quad \textbf{Equation 5.4}$$

When it comes to biases, it is very common and possible to initialize them at zero, since symmetry breaking was already achieved through the initialization of the synaptic weights [138].

At this step, a group of variables required for backpropagation called momentum terms (one momentum term for each synaptic weight) must also be initiated at zero. These variables will be explained in detail in Step 7.

Step 6: The agent takes a set of actions (equal to the mini batch size) and stores its experience in the form of [Current State, Action, Reward, Next State, Terminal State Status].

Current State is the state the agent is currently at and is stored as three images of three different views of the environment as explained in section 5.2.5.

Initially, the agent takes random actions out of the total possible 58 actions. After the simulation has performed a number of backpropagation steps greater than the "Number of Mini Batch Descents to Maximum E Greedy" as defined in Table 5.1, the agent takes actions following the SOFTMAX action selection criteria as explained in section 2.9.2.3. At this point, the agent has developed an understanding of what actions are best to take given a particular state. Therefore, it does not make sense for the agent to carry on taking actions randomly. The sequence of actions the agent takes is stored in a text file to be used in the future for backpropagation.

Each action the agent takes results in a corresponding reward value. This sequence of rewards is also stored in a separate text file for future use.

Each action the agent takes when in a particular state, changes the state of the environment to a new state. This sequence of next states is also stored as a series of 3 images each for each state for future use.

Certain actions the agent takes in a given state can result that particular learning episode to end and the environment will be reset to its initial state because of such a scenario. Such a state is called a terminal state. A list of all possible actions that can lead to a terminal state is provided below,

- Net leaving the closed environment
- Node submersion
- All 4 corner nodes of the net coming in to contact with the mould
- Nodes coming in to contact with other nodes

A record of whether each action the agent took led to a terminal state or not is stored in a text file in a binary form for future use. An action that leads to a terminal state is indexed with 1 and 0 otherwise.

Step 7: Using the experience stored by the agent to perform backpropagation steps. Once the agent has collected and stored a set of experience, this set is used to adjust the weights of the CNN through a process called backpropagation. In this section, an example provided in [141] will be used to understand the backpropagation algorithm.

For this example, a neural network with two input neurons, one hidden layer with two neurons and a bias, and an output layer with two neurons as illustrated in Figure 5.34 is used. In Figure 5.35, initial weights and biases and input/output training values are illustrated.

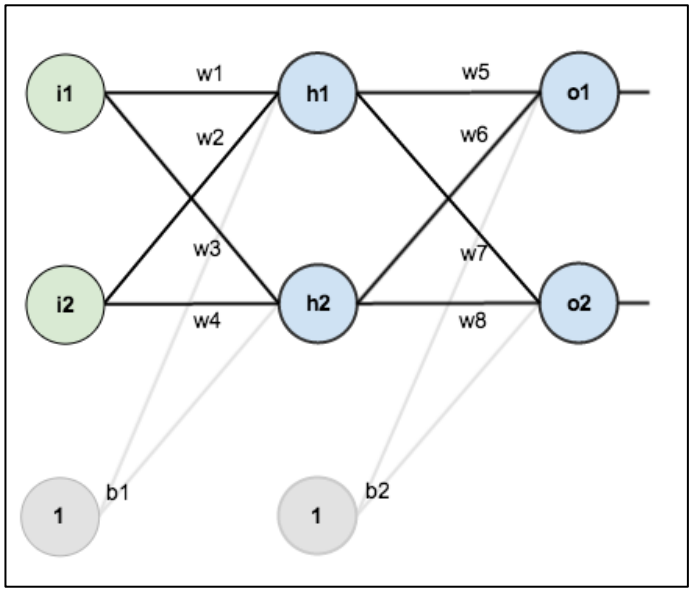


Figure 5.34 - Neural network architecture used to understand backpropagation

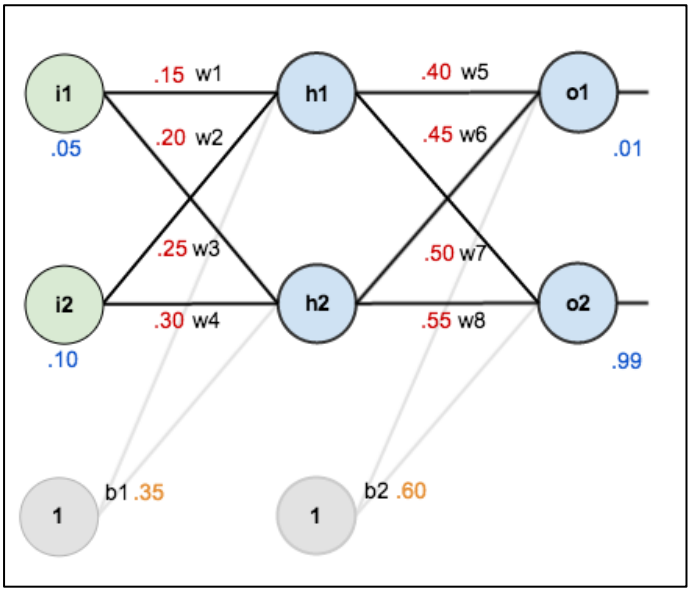


Figure 5.35 - Initial weights and biases, and training input/output variables of the neural network

The objective of backpropagation is to obtain a set of weights that will predict accurate output values for an arbitrary set of inputs. For the purpose of this example, the focus

will be given to obtaining a set of weights that will give an output vector of o = [0.01, 0.99] for an input vector of i = [0.05, 0.10].

Firstly, let us observe the output from the neural network with the initialized weights. The process of generating an output from a neural network is called a 'forward pass'.

Total input to hidden layer units:

$In_{h1} = w_1*i_1 + w_2*i_2 + b_1*1 = 0.15*0.05 + 0.2*0.1 + 0.35*1 = 0.3775$

$In_{h2} = w_3*i_1 + w_4*i_2 + b_1*1 = 0.25*0.05 + 0.3*0.1 + 0.35*1 = 0.3925$

A sigmoid activation function is then used to squash these inputs to generate outputs from these hidden units:

$Out_{h1} = \frac{1}{1+e^{-In_{h1}}} = 0.593$

$Out_{h2} = \frac{1}{1+e^{-In_{h2}}} = 0.597$

The Same process is repeated to obtain inputs and outputs of the output layer:

$In_{o1} = w_5*Out_{h1} + w_6*Out_{h2} + b_2*1 = 1.106$, $Out_{o1} = \frac{1}{1+e^{-In_{o1}}} = 0.751$

$In_{o2} = w_7*Out_{h1} + w_8*Out_{h2} + b_2*1 = 1.225$, $Out_{o2} = \frac{1}{1+e^{-In_{o2}}} = 0.773$

Outputs of the output layer can now be used to calculate the error of each output neuron and then they are summed to get the total error:

$$E_{total} = \sum \frac{1}{2} (target - output)^2 \qquad \textbf{Equation 5.5}$$

In Equation 5.5, the target refers to the true output and the output refers to the output from the neural network. A half is included so that it cancels the exponent when the total error is differentiated later. Using this equation, individual errors of output neurons can be calculated as:

$E_{o1} = 0.5\ (target_{o1} - output_{o1})^2 = 0.5(0.01 - 0.751)^2 = 0.275$

$E_{o2} = 0.5\ (target_{o2} - output_{o2})^2 = 0.5(0.99 - 0.773)^2 = 0.024$

The total error of the neural network can now be calculated as:

$E_{total} = E_{o1} + E_{o2} = 0.298$

The final step is to use these calculated errors to adjust the weights of the neural network. This process is called the backward pass. The objective of the backward pass is to update each weight of the neural network so that they cause the actual output to be closer to the target output. This should lead to a reduction of the total error of the neural network. In order to perform a single backpropagation step, we need to find the error derivatives with respect to each weight of the neural network. Consider the weights of the output layer first. For w_5, we can use the chain rule to obtain:

$$\frac{\partial \mathrm{E}_{\mathrm{total}}}{\partial \mathrm{w}_5} = \frac{\partial \mathrm{E}_{\mathrm{total}}}{\partial \mathrm{Out}_{\mathrm{o1}}} * \frac{\partial \mathrm{Out}_{\mathrm{o1}}}{\partial In_{\mathrm{o1}}} * \frac{\partial In_{\mathrm{o1}}}{\partial \mathrm{w}_5} \qquad \textbf{Equation 5.6}$$

This process is visually expressed in Figure 5.36.

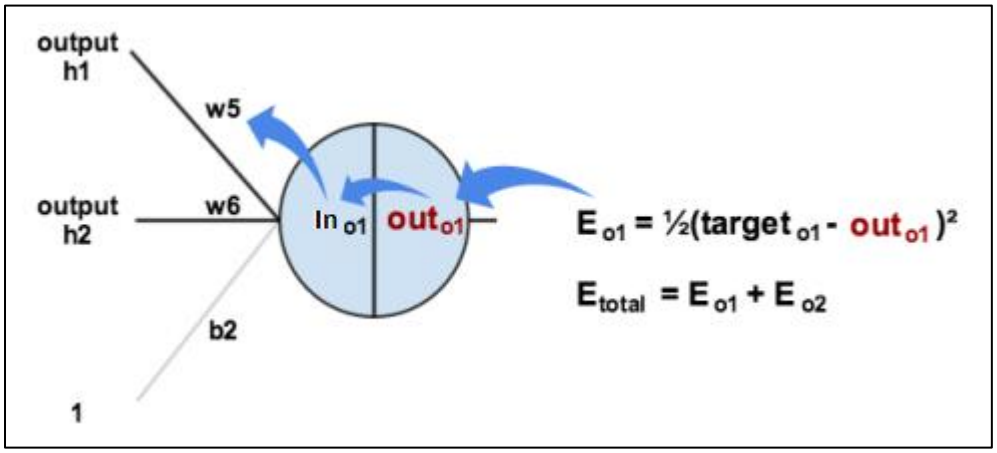


Figure 5.36 - Visual representation of the backpropagation process

Next step is to formulate each component of Equation 5.6. Firstly, consider how much the total error changes with respect to the output:

$$\mathrm{E}_{\mathrm{total}} = \frac{1}{2} * (target_{o1} - Out_{o1})^2 + \frac{1}{2} * (target_{o2} - Out_{o2})^2$$

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{o1}}} = 2 * \frac{1}{2} * (target_{o1} - Out_{o1})^{2-1} * -1 + 0 = 0.741$$

Next, consider by how much Out_{o1} changes with respect to In_{o1}:

$$Out_{o1} = \frac{1}{1+e^{-In_{o1}}}$$

$$\frac{\partial Out_{o1}}{\partial \mathrm{In_{o1}}} = Out_{o1}(1 - Out_{o1}) = 0.187$$

Finally, consider by how much In_{o1} changes with respect to w_5:

$In_{o1} = w_5 * Out_{h1} + w_6 * Out_{h2} + b_2 * 1$

$$\frac{\partial In_{o1}}{\partial \mathrm{w_5}} = Out_{h1} = 0.593$$

All these components are then combined to get:

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w_5}} = \frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{o1}}} * \frac{\partial \mathrm{Out_{o1}}}{\partial In_{\mathrm{o1}}} * \frac{\partial In_{\mathrm{o1}}}{\partial \mathrm{w_5}} = 0.741 * 0.187 * 0.593 = 0.082$$

To reduce the error of the neural network, this value is then multiplied by a learning rate (α) as mentioned in Table 5.1 and reduced from w_5:

$$w_{5\_updated} = w_5 - \alpha * \frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w_5}}$$

The same process is repeated to update w_6, w_7 and w_8. The ideal learning rate is found via trial and error. If the learning rate is too large, the error of the neural network diverges to infinity. If it is too small, a significant amount of time will be required to reduce the error of the neural network. Usually a single fixed learning rate is used to update all weights of the neural network. However, attempts have been made to use multiple adaptive learning rates to further improve the learning process ([142], [143], [144] and [145]).

Next step is to update the weights w_1, w_2, w_3 and w_4, leading to the hidden layer. Consider updating w_1, we need to figure out:

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w_1}} = \frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{h1}}} * \frac{\partial \mathrm{Out_{h1}}}{\partial In_{\mathrm{h1}}} * \frac{\partial In_{\mathrm{h1}}}{\partial \mathrm{w_1}}$$ **Equation 5.7**

A similar approach to that of updating w_5 is used here, but slightly different to account for the fact that the output of each hidden layer neuron contributes to the error of multiple output neurons. Both Out_{o1} and Out_{o2} are affected by Out_{h1}. Therefore, $\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{h1}}}$ needs to take into consideration its effect on both output neurons:

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{h1}}} = \frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{Out_{h1}}} + \frac{\partial \mathrm{E_{o2}}}{\partial \mathrm{Out_{h1}}}$$

Start with $\frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{Out_{h1}}}$:

$$\frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{Out_{h1}}} = \frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{In_{o1}}} * \frac{\partial \mathrm{In_{o1}}}{\partial \mathrm{Out_{h1}}}$$

$\frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{In_{o1}}}$ can be calculated using values calculated previously:

$$\frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{In_{o1}}} = \frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{Out_{o1}}} * \frac{\partial \mathrm{Out_{o1}}}{\partial \mathrm{In_{o1}}} = 0.741 * 0.187 = 0.138$$

And $\frac{\partial \mathrm{In_{o1}}}{\partial \mathrm{Out_{h1}}}$ is equal to w_5:

$In_{o1} = w_5 * Out_{h1} + w_6 * Out_{h2} + b_2 * 1$

$$\frac{\partial \mathrm{In_{o1}}}{\partial \mathrm{Out_{h1}}} = w_5$$

Therefore,

$$\frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{Out_{h1}}} = \frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{In_{o1}}} * \frac{\partial \mathrm{In_{o1}}}{\partial \mathrm{Out_{h1}}} = 0.138 * 0.40 = 0.055$$

The same process can be followed to get:

$$\frac{\partial \mathrm{E_{o2}}}{\partial \mathrm{Out_{h1}}} = -0.019$$

Therefore:

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{h1}}} = \frac{\partial \mathrm{E_{o1}}}{\partial \mathrm{Out_{h1}}} + \frac{\partial \mathrm{E_{o2}}}{\partial \mathrm{Out_{h1}}} = 0.055 - 0.019 = 0.036$$

Referring back to Equation 5.7, we still need to figure out two more components:

$\mathrm{Out_{h1}} = \frac{1}{1+e^{-In_{h1}}}$

$$\frac{\partial Out_{h1}}{\partial \mathrm{In_{h1}}} = Out_{h1}(1 - Out_{h1}) = 0.241$$

And,

$\mathrm{In_{h1}} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$

$$\frac{\partial In_{h1}}{\partial \mathrm{w_1}} = i_1 = 0.05$$

All previous calculations can now be put together as:

$$\frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w_1}} = \frac{\partial \mathrm{E_{total}}}{\partial \mathrm{Out_{h1}}} * \frac{\partial \mathrm{Out_{h1}}}{\partial In_{\mathrm{h1}}} * \frac{\partial In_{\mathrm{h1}}}{\partial \mathrm{w_1}} = 0.036 * 0.241 * 0.05 = 0.000439$$

w_1 can now be updated as:

$$\mathrm{w_{1\_updated}} = w_1 - \alpha * \frac{\partial \mathrm{E_{total}}}{\partial \mathrm{w_1}}$$

The same process can be repeated to update w_2, w_3 and w_4. All weights have now been updated. The total error of the neural network calculated using the updated weights must now be smaller given that an appropriate learning rate was chosen. The backpropagation algorithm explored in this example is in its most basic form and few more modifications must be made to it before it can be used in the reinforcement learning system of interest in this thesis. These modifications are discussed below.

Mini batch gradient descent: The generic gradient descent algorithm was introduced in the previous example as:

$$w_{updated} = w - \alpha * \frac{\partial E_{total}}{\partial w} \quad \textbf{Equation 5.8}$$

Where, $E_{total} = \sum \frac{1}{2} (target - output)^2$. The generic gradient descent algorithm can be further simplified for a linear regression problem with m data samples as:

$$E = \frac{1}{2m} \sum_{i=1}^{m} \left(y_t^i - y_o^i\right)^2 \quad \textbf{Equation 5.9}$$

$$w_j^{updated} = w_j - \alpha * \frac{1}{m} \sum_{i=1}^{m} \left(y_t^i - y_o^i\right).x_j^i \quad \textbf{Equation 5.10}$$

Where,

m = total number of data samples

y_t^i = true output of the i^{th} data sample

y_o^i = neural network output for the i^{th} data sample

x_j^i = j^{th} input parameter of the i^{th} data sample

In the example used above, only one data sample was present. However, in almost all cases, more than one data sample is present. Let us now consider a case where 1 million data samples are present. In this case, just to make one gradient descent step we would have to calculate each cost 1 million times if the generic form of gradient descent was used. This is very computationally expensive and will consume a significant amount of time. In addition, when applied to reinforcement learning problems, we never have access to the entire data sample at one point in time, since data samples are collected through experience. Therefore, an alternative approach is required. There are two different approaches to solve this issue.

1. Stochastic gradient descent (SGD): In stochastic gradient descent, one data sample is used at each iteration. This is identical to the example presented above. Because we use one data sample in SGD, the path to the minimum cost point tends to be very noisy and there is a very high probability that the agent will fail to even get close to this minimum, since a single data sample fails to represent the entire state of the environment.

2. Mini batch gradient descent: This is the preferred method in training neural networks that contain a large number of data samples. In mini batch gradient descent, n data samples are used to perform one gradient descent iteration instead of 1. The ideal value for n is usually found by trial and error. n should be large enough such that each mini batch contains sufficient information to represent the behaviour of the environment. The momentum method [146] has been used to optimize the process of mini batch gradient descent in the simulation presented in this thesis. The momentum method greatly improves the speed of learning of a neural network by accelerating the gradient vectors in the right directions and dampening oscillations:

$$v_t = 0.9 * v_{t-1} + (1 - 0.9)\alpha \frac{\partial \mathrm{E}}{\partial \mathrm{w}} \qquad \textbf{Equation 5.11}$$

$$w^{updated} = w - v_t$$

Data pre-processing: As explained earlier, states of the reinforcement learning system, which also become the input to the CNN, are represented by images. An image can be represented by its RGB values numerically. RGB values range from 0 to 1 with a mean value of 0.5. This mean value is deducted from each RGB value to zero centre the input data. It is common practice to use zero centred data to increase the learning speed of CNNs [138].

Target values of the agent: Target values of the agent are defined based on the terminal state value. If the terminal state value is 0, the target value for an action taken by the agent at a particular state is the sum of the instantaneous reward received and the discounted maximum action value of the next state. If the terminal state value is 1, the target value is just the instantaneous reward.

With this understanding, the sequence of steps taken to perform a single backpropagation step is in the simulation including initialization of relevant parameters can be summarised as:

- Initialize action-value function Q with random weights w
- Initialize target action-value function Q* with weights w^* = w
- Initialize momentum values v to zero

For Mini Batch = 1 to Infinity (or large value)

- Acquire data sample [Current State, Action, Reward, Next State, Terminal State Status] of size equal to the Mini Batch Size.
- Initialize cost E_{total} to zero
- Initialize $\frac{\partial E}{\partial w_{total}}$ = 0

For Integer ii=1 to ii=Mini Batch Size

- Zero centre $[Current\ State]_{ii}$
- Zero centre $[Next\ State]_{ii}$
- Calculate action values for $[Current\ State]_{ii}$ using Q with weights w
- Calculate action values for $[Next\ State]_{ii}$ using Q^* with weights w^*
- **If** $[Terminal\ State\ Status]_{ii}$ = 0
  - Target value for $[Action]_{ii}$ = $[Reward]_{ii}$ + Discount Rate(ϒ) * maximum of (action values for $[Next\ State]_{ii}$)

  Else
  - Target value for $[Action]_{ii}$ = $[Reward]_{ii}$

  End If
- Set [Target Action Values] = action values for $[Current\ State]_{ii}$
- Set [Target Action Values] ($[Action]_{ii}$)= Target value for $[Action]_{ii}$
- Now, referring to Equation 5.5, output = action values for $[Current\ State]_{ii}$ using Q with weights w and target = [Target Action Values]
- $E_{total} = E_{total} + sum(0.5*(target-output)^2)$

- Perform gradient descent step with respect to network parameters w and acquire $\frac{\partial \mathrm{E}}{\partial \mathrm{w}_{ii}}$
- $\frac{\partial \mathrm{E}}{\partial \mathrm{w}_{total}} = \frac{\partial \mathrm{E}}{\partial \mathrm{w}_{total}} + \frac{\partial \mathrm{E}}{\partial \mathrm{w}_{ii}}$

End For

- $\mathrm{E} = \mathrm{E}_{\mathrm{total}}$ / (Mini Batch Size)
- $\frac{\partial \mathrm{E}}{\partial \mathrm{w}} = \frac{\partial \mathrm{E}}{\partial \mathrm{w}_{total}}$ / (Mini Batch Size)
- $v_t = 0.9 * v_{t-1} + \alpha \frac{\partial \mathrm{E}}{\partial \mathrm{w}}$
- $w = w - v_t$
- After every C steps set w^* = w
- Increment mini batch size by a constant value (optional)

End For

The entire backpropagation step is performed in MATLAB (R2017a). An inbuilt toolbox in MATLAB for convolutional neural networks that exploits GPU computing is used to calculate $\frac{\partial \mathrm{E}}{\partial \mathrm{w}_{ii}}$.

A mini batch of experience is collected in Unity 3D and the simulation is then paused until a single gradient descent step is performed in MATLAB. This process is then repeated until the AI agent learns a sequence of actions that yields a maximum total reward.

5.3 Enhancements to the Reinforcement Learning System

In section 5.2, all basic components of the reinforcement learning system have been explained. However, few additional components were used to optimize and enhance the learning process of the AI agent. These components are discussed in this section.

5.3.1 Gradual Increase of the Mini Batch Size

As stated in Table 5.1, the 'mini batch size' hyper-parameter is initialized at 100 and the 'minimum number of episodes per mini batch' is fixed at 10. This means that the 'maximum number of actions per episode' is 10 (100/10). However, to complete a particular layup, more than 10 actions may be required depending on the mould geometry and the net resolution. Therefore, the mini batch size must be increased within the simulation if the agent is to arrive at a final solution. A straightforward solution to this is to start the simulation at a very large mini batch size and keep it constant. However, gradient descent steps are highly computationally expensive and hence the mini batch size must be kept as small as possible to speed up the learning process. Therefore, initially the mini batch size is kept low and it is incremented by a constant amount after each gradient descent step. Due to the small mini batch size at the beginning of the simulation, the learning process is accelerated and it will be proved in the next chapter that a small mini batch size is sufficient for the agent to create a good enough representation of the behaviour of the environment at the early learning stages. Therefore, fine-tuning of the ideal sequence of actions required for a particular layup is done at the later stages of the simulation with large mini batch sizes.

5.3.2 Forced Multiple Episodes per Mini Batch Learning

By introducing the hyper-parameter 'minimum number of episodes per mini batch', each mini batch used for gradient descent is forced to have experience ranging over multiple episodes. This is essential to ensure that each mini batch is representative of the behaviour of the environment and hence, making the convergence to the minimum faster.

5.3.3 Long Term Memory of the AI Agent

The AI agent is given the ability to remember sequences of actions that yielded high total rewards to explore further in the future. The number of such sequences the agent can remember is controlled through the hyper-parameter 'memory capacity'.

For the purpose of further understanding this concept, let us assume an agent with 'memory capacity' 3 and a setting of 'minimum number of episodes per mini batch' equal to 10. In such a situation, the agent can store a maximum of 3 sequences of actions that yield the highest cumulative rewards. At the beginning of the simulation, all three memory slots are empty. As the agent explores the environment randomly, it is bound to find a sequence of actions that yields a cumulative reward greater than 0 at the end of an episode. The agent immediately stores this sequence of actions omitting any null actions. The remaining 2 memory slots are filled in a similar manner. Once all 3 memory slots are full, there are two scenarios in which the AI agent can replace one of them:

1. If in the future, the AI agent comes across any sequence of actions that yields a higher cumulative reward than the sequence of actions that yields the minimum cumulative reward among its memory capacity.

2. If in the future, the AI agent comes across any sequence of actions that yields the same cumulative reward as the sequence of actions that yields the minimum cumulative reward among its memory capacity, but yields this cumulative reward in a lesser number of actions.

While one or more memory slots are filled with a sequence of actions at the beginning of the creation of a new mini batch, the agent always explores all of its stored action sequences furthermore to check if the current respective cumulative reward can be further increased. If the agent succeeds in further increasing this cumulative reward, it immediately updates its memory. Therefore, when all 3 memory slots are full, 3 out of the 10 'minimum number of episodes per mini batch' will always be a further exploration of stored actions sequences that yields high rewards. This greatly increases the chance of the agent in finding the most optimal sequence of actions to layup a particular mould shape.

Each time the agent updates its memory; all sequences of actions in its memory are written to an output text file ranking them from the highest cumulative reward to the lowest. This is the main output file from the simulation. These action sequences can then

be compared with each other at the end of the simulation to determine the ideal layup sequence for a particular composite part.

5.3.4 Optimization of Long Term Memory of the AI Agent

It was observed during simulations that the agent is capable of taking actions that cancel out with each other and hence does not contribute to the completion of the layup. An example of such a layup sequence is provided below.

Action Sequence: Action 50**, *Action 36, Action 34***, Action 34, Action 34

It is clear from Solution Path 1, as explained in section 6.2.1, that the two actions in bold letters in the above action sequence are not required to complete the layup. These two actions cancel each other in this case (Refer to Figure 5.16 and Figure 5.17). However, the final cumulative reward is unchanged since no negative rewards are received for performing these two actions. Therefore, in such cases, before saving this sequence of actions in to the agent's long-term memory, it must be optimized to ignore such cancelling pairs of actions. This particular optimization is carried out by repeating the sequence of actions for every possible combination of omissions of actions and finding the sequence of actions that yield the highest reward at the lowest possible number of actions. All possible omission patterns are tabulated in Appendix: Table 3 for this particular sequence of actions.

5.4 Summary

In this chapter, the computational modelling process in creating a reinforcement-learning environment with an AI agent has been discussed. All six main elements of the reinforcement learning system has been discussed in detail. The closed environment and the agent-environment interactions have been modelled in Unity 3D as a video game. A CAD model of the composite part to be laid up is imported in to the Unity 3D and the woven reinforcement is modelled as a rigid pin jointed net. The agent is given the ability to perform a fixed number of actions in the form of manipulations to the net. A reward system has been put in place that awards the agent with positive and negative rewards based on actions taken by the agent. The goal of the agent is to maximize the

total cumulative reward. The agent is programmed to accumulate a mini batch of experience from the Unity 3D simulation and then transfer this batch of experience to MATLAB to perform a single gradient descent step that allows the agent to learn the behaviour of the environment. The core of the agent is a convolutional neural network that approximates the action values of all actions the agent can take at a given state. Further enhancements such as gradual increase of mini batch size, forced multiple episodes per mini batch and long-term memory of the AI agent were introduced to speed up and enhance the learning process of the AI agent. In the next chapter, a detailed analysis of the computational model and novel optimization methods introduced in this chapter will be presented.

CHAPTER 6: An Artificially Intelligent Layup Agent: Preliminary Case Study (Results)

6.1 Introduction

In the previous chapter, the computational model for an artificially intelligent layup agent capable of learning the best sequence of actions required to layup composite parts was introduced. A simple mould shape with a flat surface followed by a ramp at 45^{o} was used to explain the modelling process. In this chapter, the same mould shape and computational model are used to obtain and discuss results of such a simulation. The behavioural changes of the simulation with respect to hyper-parameters listed in Table 5.1 are also explored.

6.2 Exploring Solution Paths to the Preliminary Case Study

With the mould shape and net resolution chosen for the preliminary case study, there are three possible solution paths,

1. Action 50 (Down), Action 34 and Action 34
2. Action 34, Action 34, and Action 50 (Down)
3. Action 34, Action 50 (Down) and Action 34

These solution paths are explained in more detail below:

6.2.1 Solution Path 1

Action sequence required for solution path 1: Action 50 (Down), Action 34 and Action 34. This is illustrated in Figure 6.1. The net is first translated down to secure it on to the flat surface, and then two out of plane straight bending actions are performed to complete the layup.

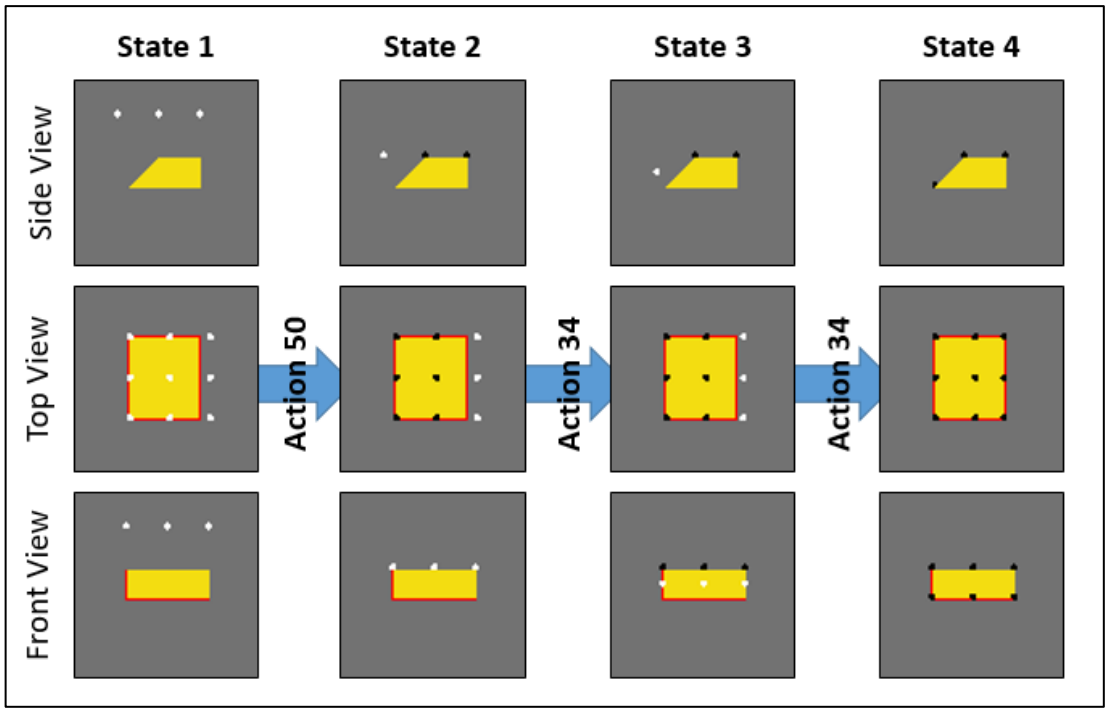


Figure 6.1 - Action sequence for solution path 1

6.2.2 Solution Path 2

Action sequence required for solution path 2: Action 34, Action 34, and Action 50 (Down). This is illustrated in Figure 6.2. The net is first bent out of plane twice (by 45° in total) to match the ramp angle and then stuck on to the face of the mould by translating it downwards.

Both solution path 1 and 2 yield the same final cumulative reward and require three actions to complete the layup. However, in solution path 1, the agent will receive an immediate reward after performing the first action, since six nodes come in to contact with the mould, in contrast to solution path 2, where the agent will only receive a reward upon performing the third action. Therefore, the agent will converge to this solution during the learning process rather than the second solution path. This will be proved in section 6.3.

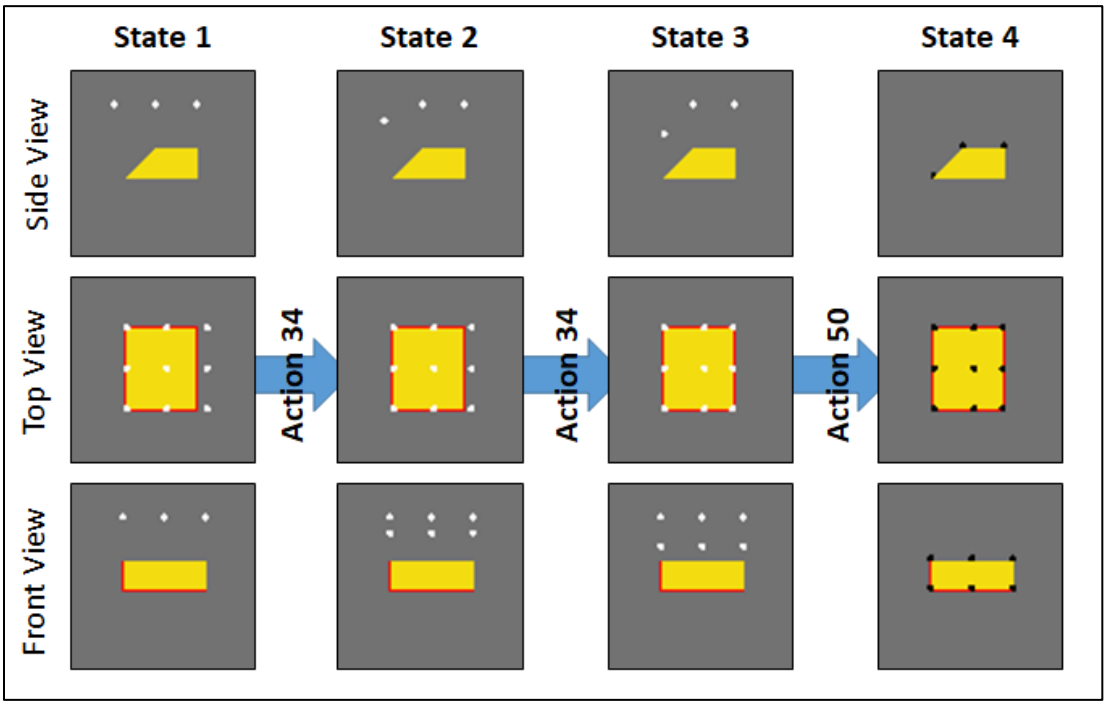


Figure 6.2 - Action sequence for solution path 2

6.2.3 Solution Path 3

Action sequence required for solution path 3: Action 34, Action 50 (Down) and Action 34. This is illustrated in Figure 6.3. The net is first bent out of plane by 22.5°, then translated downwards and finally bent by another 22.5° to secure it on to the ramp.

Solution path 3 yields the same cumulative future reward as solution paths 1 and 2. The agent will prefer solution path 1 to 3, since solution path 1 yields a more immediate reward. State 2 of both solution paths 2 and 3 are the same. When the agent is at this state, it will prefer solution path 3 to 2, since it will receive an immediate reward by translating the net downward, rather than bending it out of plane. This will also be proved in section 6.3.

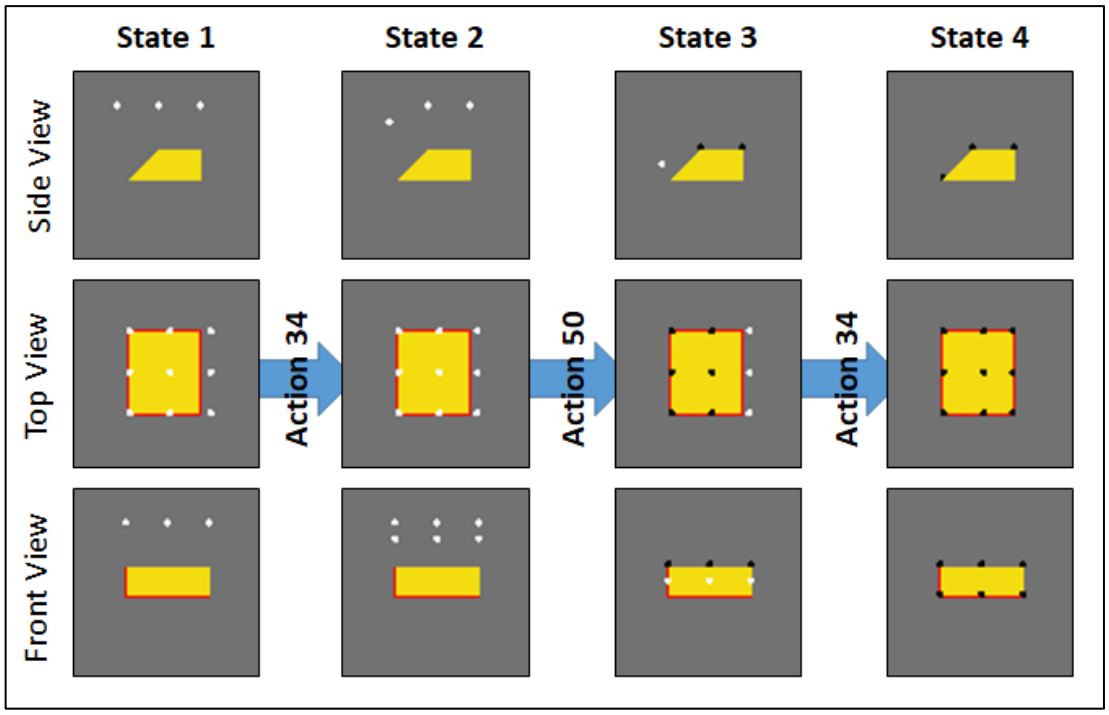


Figure 6.3 - Action sequence for solution path 3

6.3 Proof of the Reinforcement Learning Algorithm

In section 6.2, all possible solution paths and even the ideal solution path (solution path 1) to the preliminary case study have been established prior to running the simulation. This was only possible due to the simplicity of the mould shape. For more complex shapes, we cannot make such conclusions. We will now exploit this knowledge to prove that the simulation is in fact accurate and the agent actually is capable of figuring out on its own what the ideal layup path for a particular mould shape is. In order to achieve this, action value predictions for states 1, 2 and 3 from solution path 1 (see Figure 6.1) using Q with weights 'w' are plotted against number of gradient descent iterations. For the purpose of understanding the results, these 3 states are named Free Net State, Net on Mould State and Net on Mould Bent State respectively. Hyper-parameters used for this simulation are shown in Table 5.1. The mini batch size was kept constant though out the simulation since only three steps are required to complete the layup of this particular mould shape.

It is worth recalling at this point that the action value prediction for a particular action at a given state corresponds to the maximum expected cumulative reward of taking that particular action at that particular state. For the convenience of the reader, three time stamps (18, 40 and 1800 gradient descent iterations) were chosen to study the evolution of the agent. Figure 6.4, Figure 6.5, and Figure 6.6 illustrate the action value predictions at the initial learning stage (up to 18 gradient descent iterations). It is clear from Figure 6.4 that after just three iterations, the agent has determined that the action 'Down' is the most favourable action to take at the Free Net State. This notion is further solidified at the end of 18 iterations as the action 'Down' is firmly placed above all other actions with the gradient of its graph close to zero. However, action 'Down' is still ranked very high for states Net on Mould and Net on Mould Bent as evident from Figure 6.5 and Figure 6.6. For these two states, the action 'Down' is a null action and yields a negative reward. Therefore, the agent has not yet succeeded in fully distinguishing between these states. However, the slope of action 'Down' is clearly negative at the end of 18 iterations for both these states and this is an indication that the agent is heading towards ranking the action 'Down' where appropriate. While Action 34 is highlighted in all three Figures, not enough iterations have been made to come to any conclusion on the agent's understanding of Action 34.

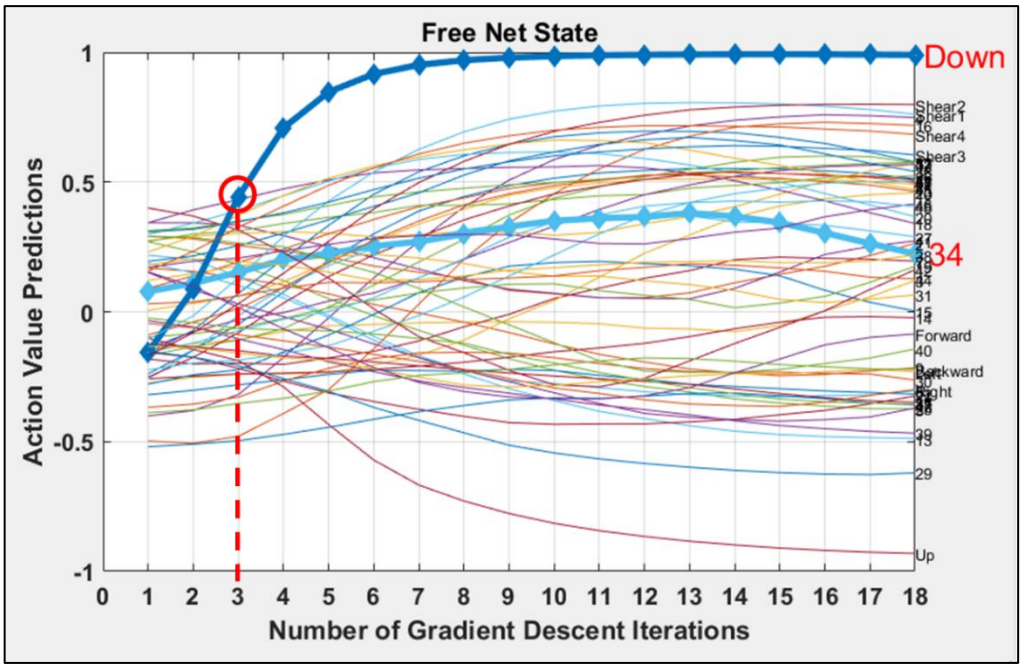


Figure 6.4 - Action value predictions for the 'free net state' up to 18 gradient descent iterations

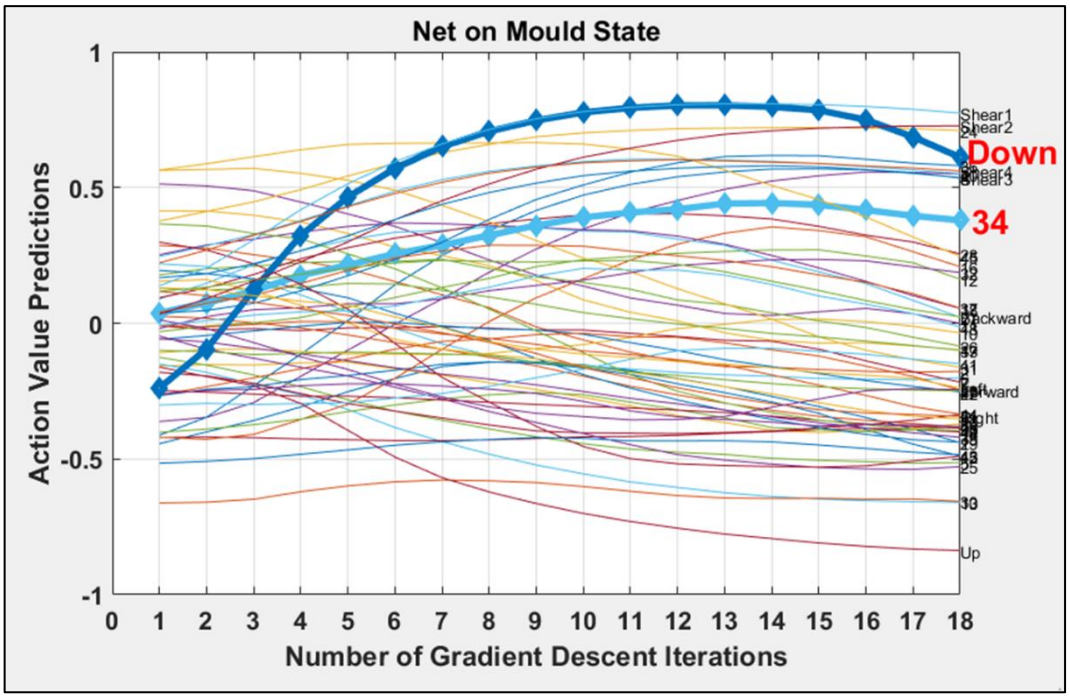


Figure 6.5 - Action value predictions for the 'net on mould state' up to 18 gradient descent iterations

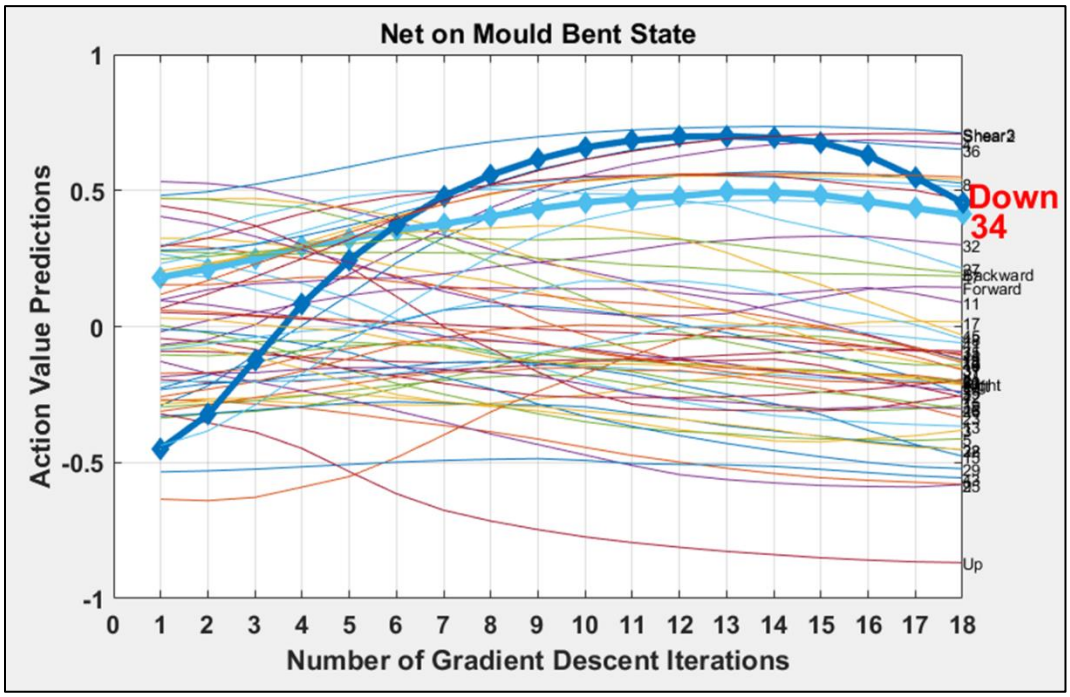


Figure 6.6 - Action value predictions for the 'net on mould bent state' up to 18 gradient descent iterations

Action value predictions for the same three states are now observed at 40 learning iterations. This is illustrated in Figure 6.7, Figure 6.8, and Figure 6.9. Not much has changed for the Free Net State at 40 iterations when compared to at 18 iterations as evident from Figure 6.7. Action 'Down' is still ranked at the top. However, for Net on Mould State and Net on Mould Bent State, the action 'Down' is now ranked at the bottom appropriately as evident from Figure 6.8 and Figure 6.9. The agent has also identified that Action 34 is critical in completing the layup by ranking it in the top half.

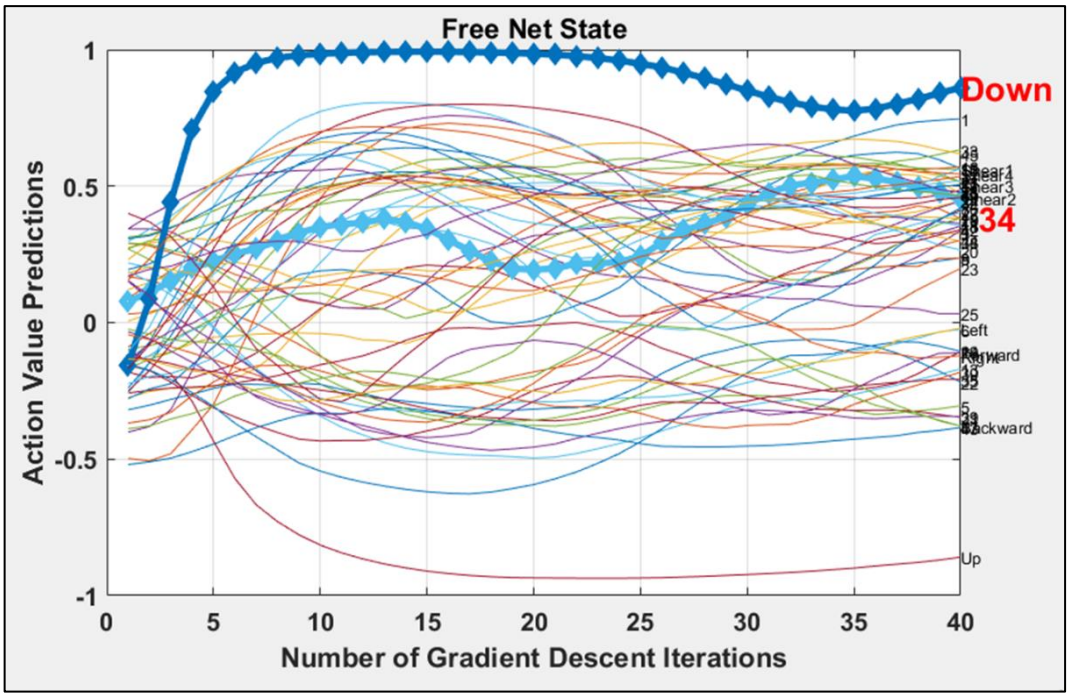


Figure 6.7 - Action value predictions for the free net state up to 40 gradient descent iterations

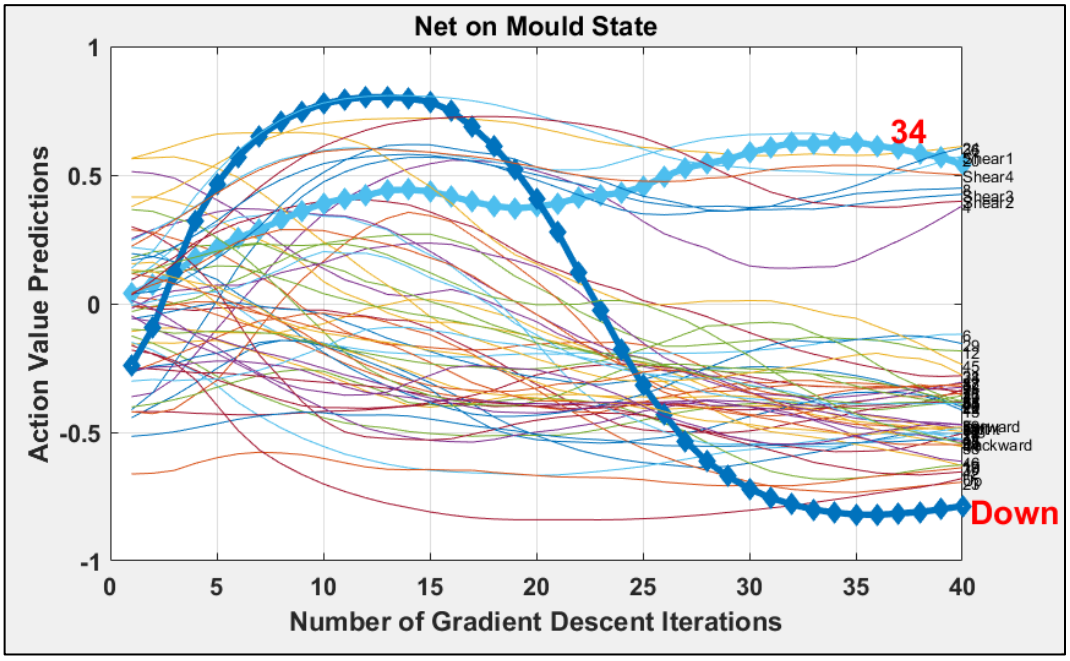


Figure 6.8 - Action value predictions for the 'net on mould state' up to 40 gradient descent iterations

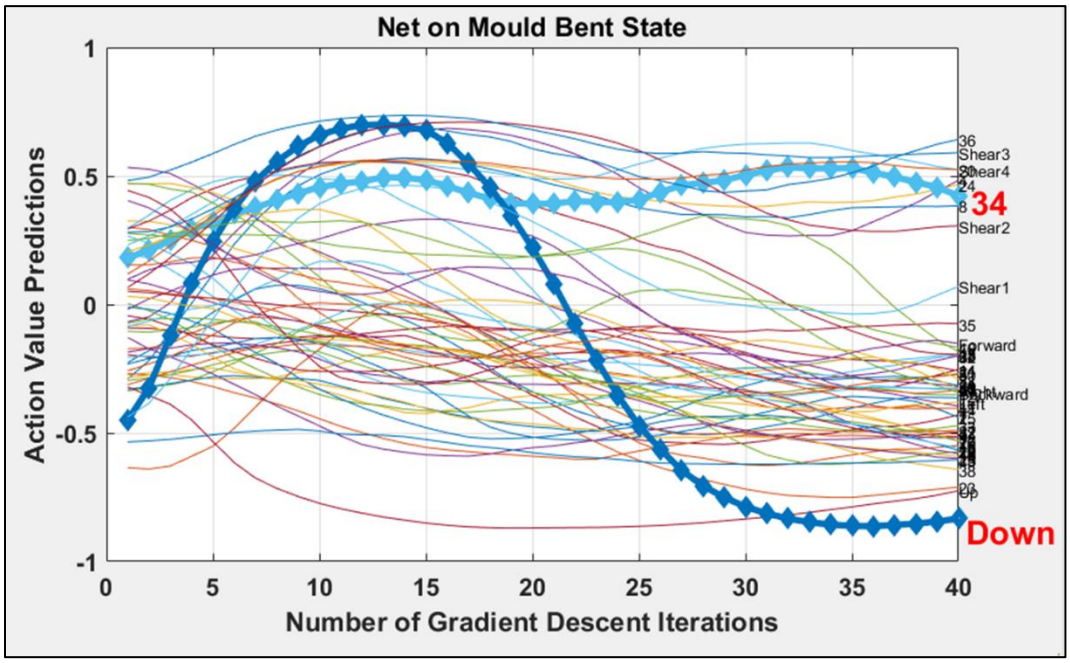


Figure 6.9 - Action value predictions for the' net on mould bent state' up to 40 gradient descent iterations

The simulation was then left to run overnight to obtain matured results. These results are illustrated in **Error! Reference source not found.**, Figure 6.11, and Figure 6.12. As evident from **Error! Reference source not found.**, action 'Down' is still ranked at the top for the Free Net State. The sudden step change observed at around 300 iterations (marked by a red ellipse) is due to setting $w^* = w$ at C steps as explained in section 5.2.6 (here C is 300, as stated in Table 5.1).

It is also clear that the agent has divided all 58 actions in to two distinct groups. All actions that result in a future cumulative negative reward are grouped at the bottom and the rest of the actions that result in a future cumulative positive reward are grouped at the top. A similar grouping can be observed for the remaining two states as evident from Figure 6.11 and Figure 6.12, with 'Action 34' being included in the top group. This grouping means that, when the SOFTMAX action selection criteria is applied, the agent is more likely to take actions that yield a high positive future cumulative reward at each state, thereby making the learning process more efficient.

It is also worth noticing that 'Action 34' is not ranked at the top for the Net on Mould Bent state. This is because the objective of the backpropagation algorithm is to find a solution to the reinforcement learning system that results in a global minimum error. Since, 'Action 34' is associated with high negative rewards of '-1' to avoid wrinkling due to bending-bending coupling, in order to minimize the global error, its action value has not been placed at the top for this particular state. This is a drawback in determining the ideal action sequence. Since, 100% accuracy cannot be achieved in mapping states to their action values, a certain element of 'luck' is needed for the agent to find the ideal action sequence within a finite period of time.

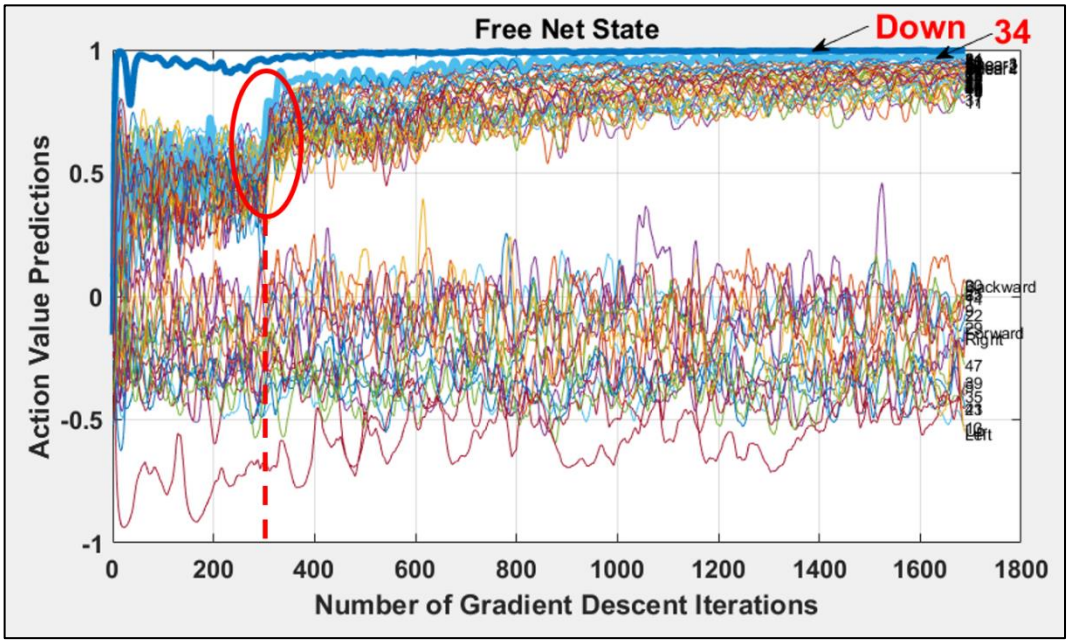


Figure 6.10 - Action value predictions for the 'free net state' up to 1700 gradient descent iterations

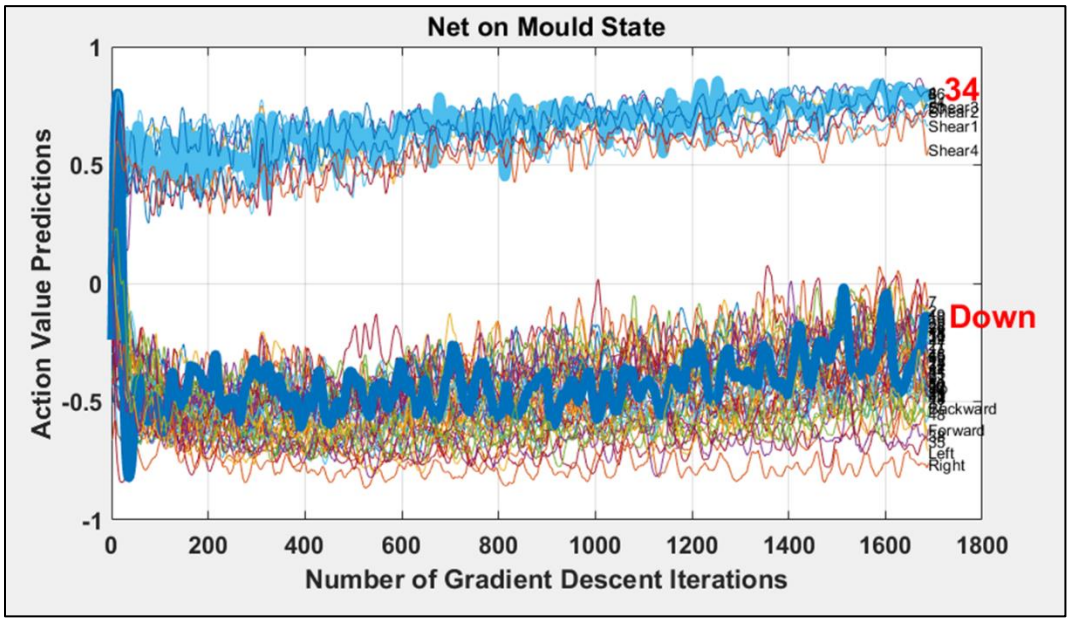


Figure 6.11 - Action value predictions for the 'net on mould state' up to 1700 gradient descent iterations

Note: In Figure 6.11 and Figure 6.12, the thicker light blue line corresponds to 'Action 34', and the thicker dark blue line corresponds to 'Action Down'.

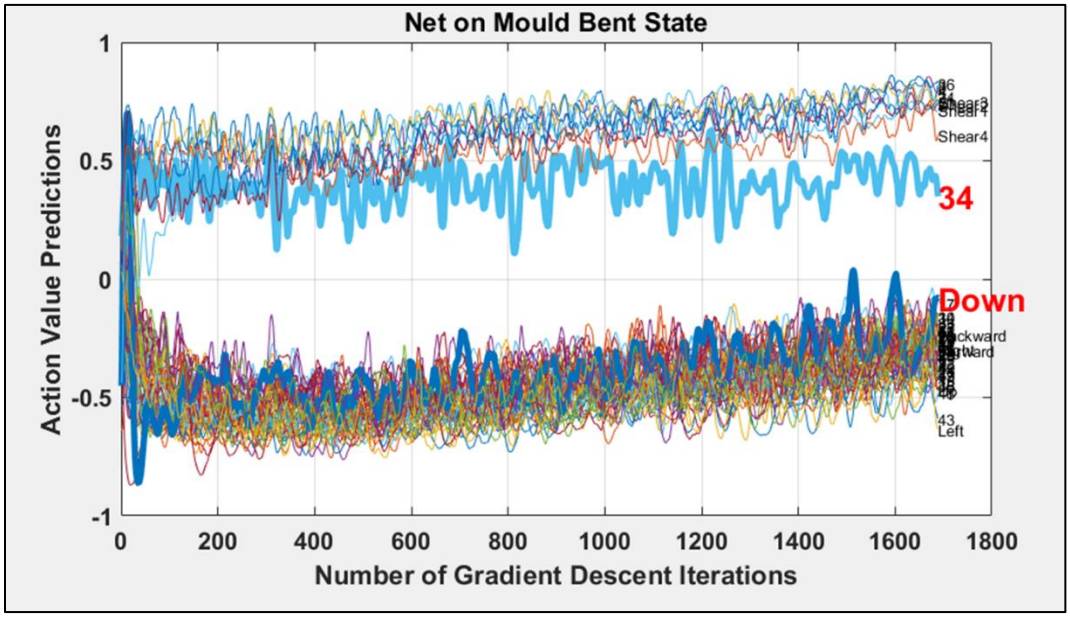


Figure 6.12 - Action value predictions for the 'net on mould bent state' up to 1700 gradient descent iterations

Figure 6.13 illustrates the action value predictions for the Free Net Bent State. The action 'Down' is ranked above 'Action 34' in this case. This is proof that the agent prefers solution path 3 to solution path 2 as stated in section 6.2.3.

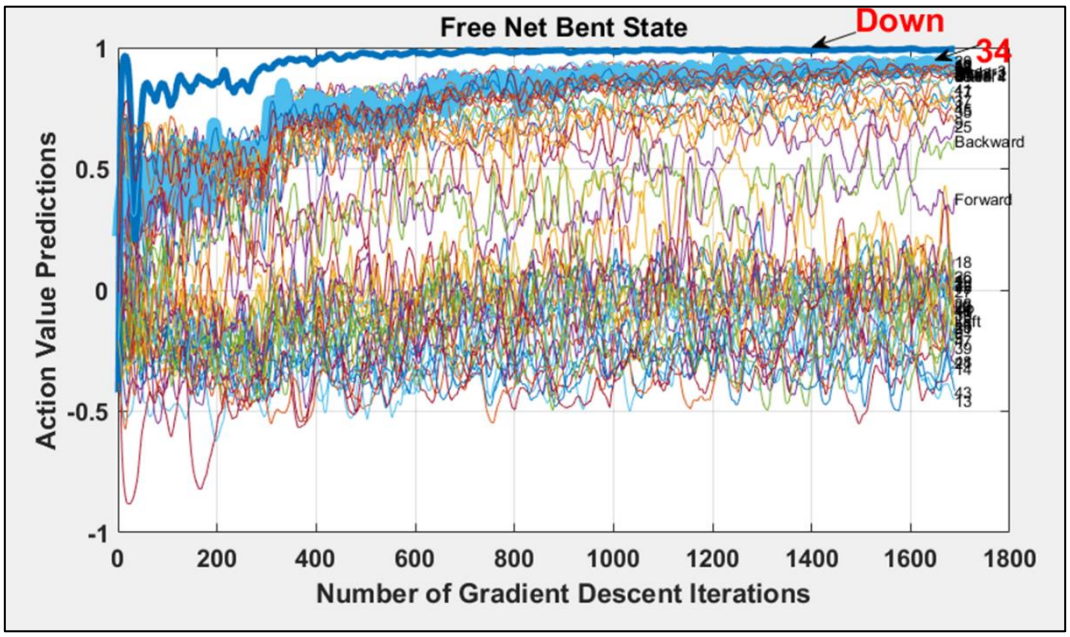


Figure 6.13 - Action value predictions for the 'free net bent state' up to 1700 gradient descent iterations

6.4 Hyper-Parameter Study

In this section, the effect of hyper-parameters: learning rate, mini batch size, number of minimum episodes per mini batch, weight swap steps, rotational constant, translational step and shear deformation step, number of mini batch gradient descents to maximum E greedy and memory capacity on the simulation will be explained.

6.4.1 Effect of the Learning Rate on the Simulation

As mentioned earlier, the ideal learning rate for a particular CNN structure is found via trial and error. Once the ideal learning rate is found, it becomes evident from the cost function of the CNN when plotted against the number of gradient descent iterations. The cost should gradually decrease and settle down at a negligible value. Figure 6.14 illustrates this trend when a learning rate of 0.01 is used. If the chosen learning rate is too high, the cost usually increases and settles down at a larger value. This is illustrated

in Figure 6.15, with the simulation run at a learning rate of 10 with all other hyper-parameters unchanged.

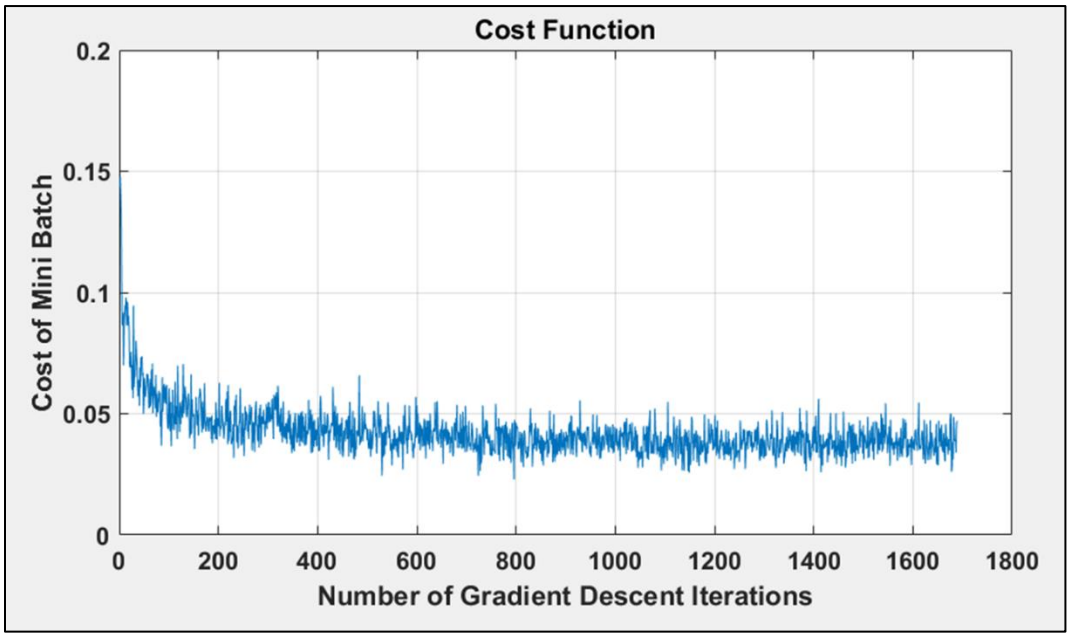


Figure 6.14 - Cost function of the convolutional neural network with a learning rate of 0.01

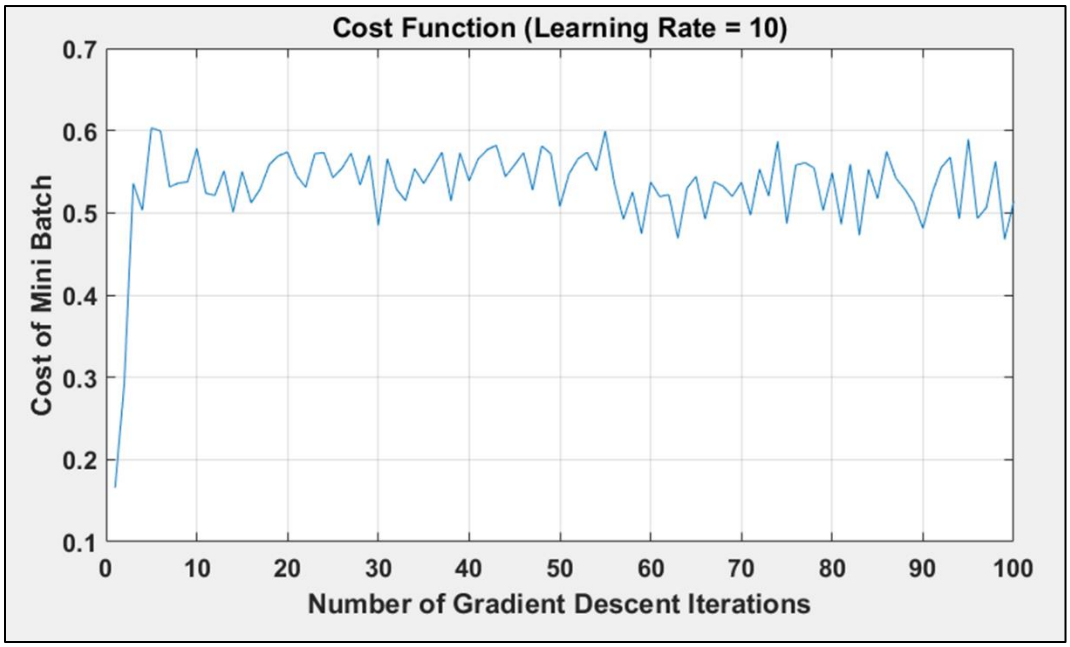


Figure 6.15 - Cost function of the convolutional neural network with a learning rate of 10

When such a large learning rate is used, the agent is unable to learn an accurate representation of the environment. In such a case, action value predictions converge to

maximum and minimum values of +1 and -1, as illustrated in Figure 6.16, which represents the Free Net State. Identical results are obtained for all other states.

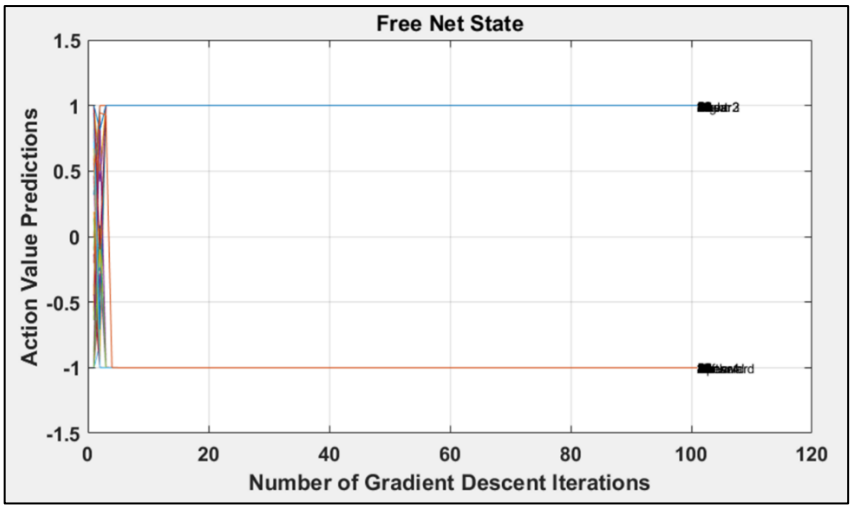


Figure 6.16 - Action value predictions for the 'free net state' with a learning rate of 10

If the chosen learning rate is small compared to the ideal learning rate, it will take a higher amount of iterations for the cost to reduce as illustrated in Figure 6.17, where a learning rate of 0.0001 is used. At 100 learning iterations, the cost is above 0.1, whereas when a learning rate of 0.1 was used, the cost was close to 0.05 at 100 iterations as illustrated in Figure 6.14.

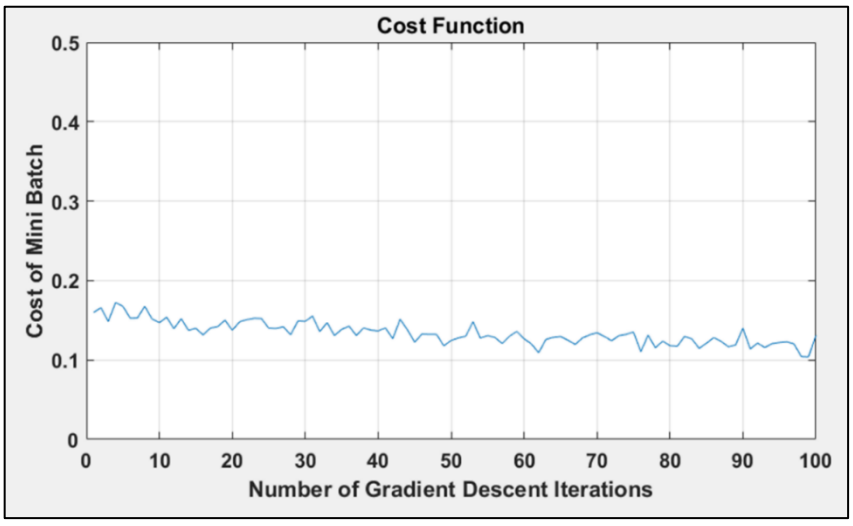


Figure 6.17 - Cost function of the convolutional neural network with a learning rate of 0.0001

Another result of this slow learning rate is that the agent will take a long time to develop a representation of the behaviour of the reinforcement-learning environment. Action value predictions for the Net on Mould State after 100 learning iterations with a learning rate of 0.0001 are illustrated in Figure 6.18. In the case where a learning rate of 0.01 was used, just after 40 learning iterations, the agent managed to divide all 58 actions in to two groups and identify which actions were more favourable compared to others as illustrated in Figure 6.8. The agent also managed to identify the action 'Down' as a null action for this particular state. However, in this case, when the learning rate has been reduced to 0.0001, even after 100 iterations, the agent has failed to identify this null action. Therefore, identifying the ideal learning rate is crucial to the success of this simulation.

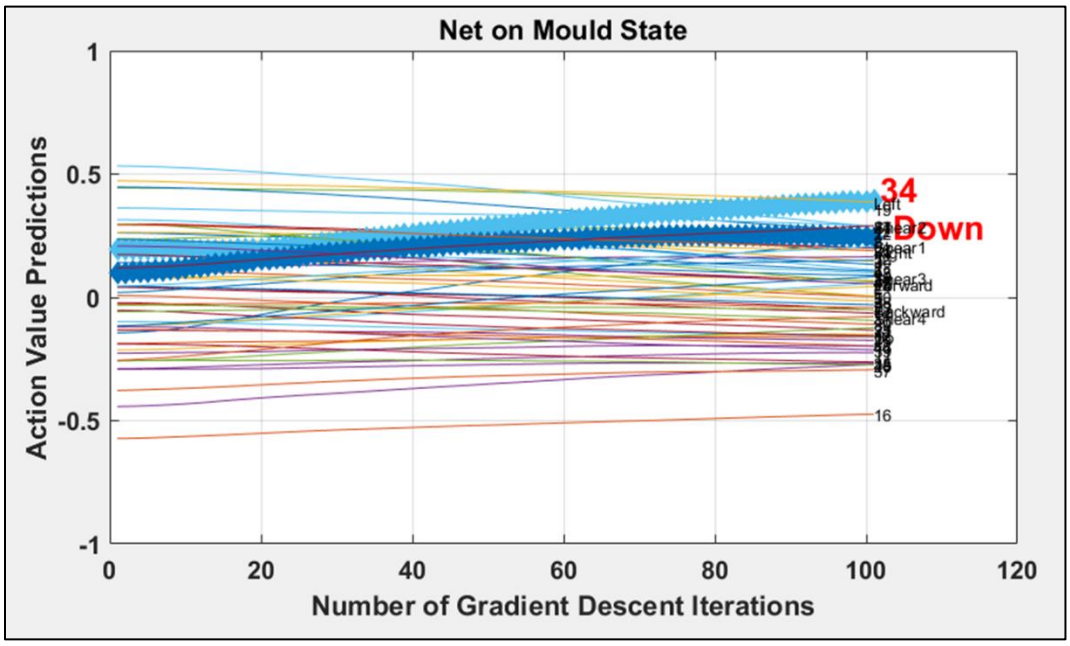


Figure 6.18 - Action value predictions for the 'net on mould state' with a learning rate of 0.0001

6.4.2 Effect of the Mini Batch Size on the Simulation

The concept of mini batches had to be introduced since when it comes to reinforcement learning systems, at a single point in time, the agent does not have access to the entire data set required for training and it is computationally expensive to perform gradient descents on the entire data set when it is very large. Ideally, what is required is the

smallest mini batch size that is representative of the behaviour of the environment, which will not hinder the learning process. Increasing the mini batch size will always smoothen the learning process and increase its accuracy. Therefore, the goal is to find the smallest possible size of the mini batch.

To understand the behaviour of the simulation at very small mini batch sizes, the simulation was run at a mini batch size of 20, number of minimum episodes per mini batch of 2, and a memory capacity of 1 with other hyper-parameters unchanged. It is clear from Figure 6.19 that even though there is a global trend of decreasing cost, there are significant oscillations of it while it decreases. These oscillations affect the agent's understanding of the behaviour of the environment as illustrated in Figure 6.20. Even after close to 650 iterations, the agent has failed to identify that action 'Down' is a null action for the Net on Mould State. In addition, the agent has ranked 'Action 34' at the bottom, showing that it has failed to identify that this particular action is critical to completing the layup. In contrast, when a larger mini batch size was used as illustrated in Figure 6.8, the agent managed to develop a much better understanding of the behaviour of the environment after a very short number of learning iterations. Therefore, it is crucial to use a mini batch size that is large enough for the success of the simulation.

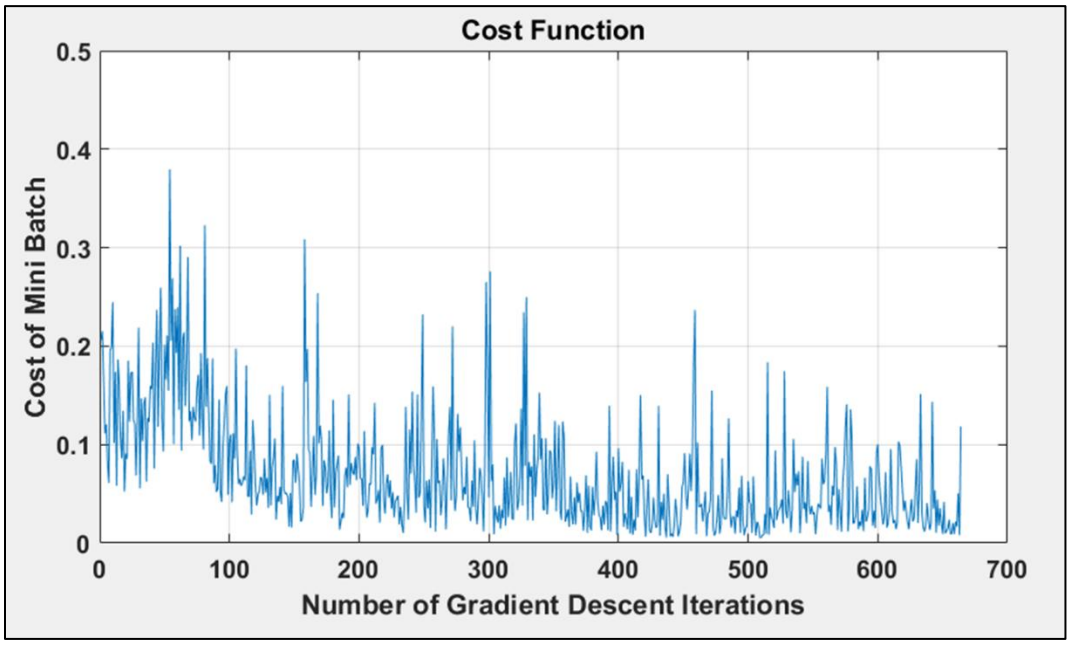


Figure 6.19 – Cost function of the convolutional neural network with a mini batch size of 20, number of minimum episodes per mini batch of 2 and a memory capacity of 1

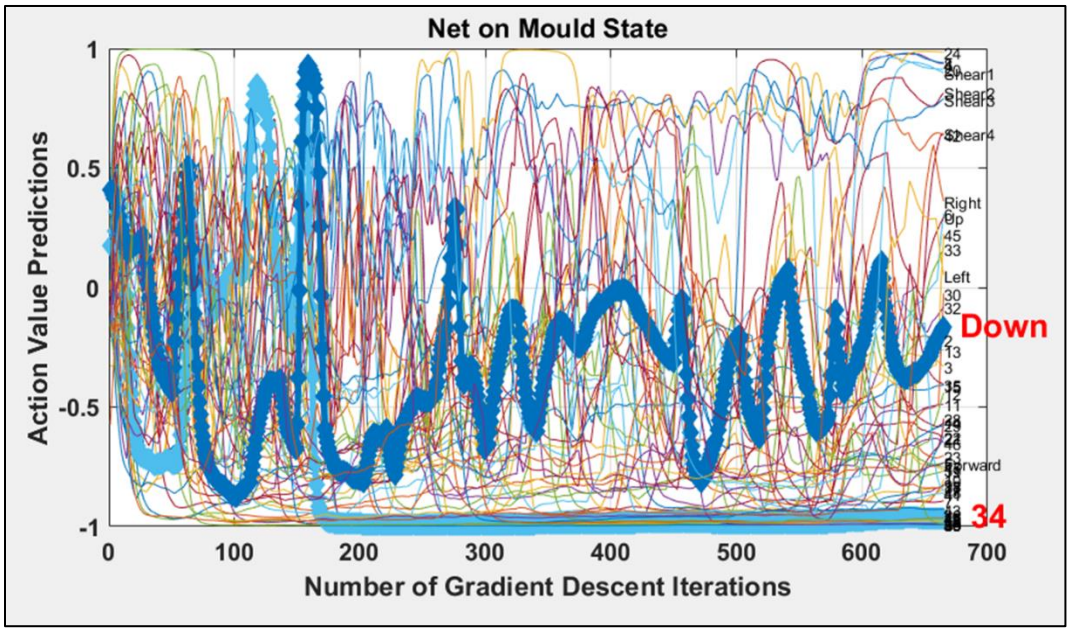


Figure 6.20 - Action value predictions for the 'net on mould state' with a mini batch size of 20, number of minimum episodes per mini batch of 2 and a memory capacity of 1

6.4.3 Effect of the Number of Minimum Episodes per Mini Batch on the Simulation

The importance of having a large enough mini batch size was explained in section 6.4.2. While the size of the mini batch is critical for the success of the simulation, it is equally critical to ensure that each mini batch is representative of the behaviour of the environment. For this particular simulation, this was achieved by forcing each mini batch to contain experience from multiple episodes as explained in section 5.3.2. Larger the number of episodes per mini batch, the more representative each mini batch becomes of the environment.

The simulation used in section 6.4.2 is also an example for a case where a small number of minimum episodes per mini batch is used. To highlight the importance of having a large enough number of episodes per mini batch, the same simulation was run by increasing this number from two to five.

Results of this simulation are illustrated in Figure 6.21 and Figure 6.22. Significant amount of oscillation can still be observed in Figure 6.21. However, it is certainly less intense than what was observed in Figure 6.19. However, an increase of the number of minimum episodes per mini batch from two to five has significantly improved the agent's understanding of the behaviour of the environment. This is evident from Figure 6.22, since the agent has now determined that action 'Down' is a null action for the Net on Mould State and the agent has understood the importance of Action 34, since it is ranked at the top. Therefore, it is critical to ensure that each mini batch contains sufficient information about the behaviour of the environment by forcing each mini batch to contain experience from a sufficient number of episodes.

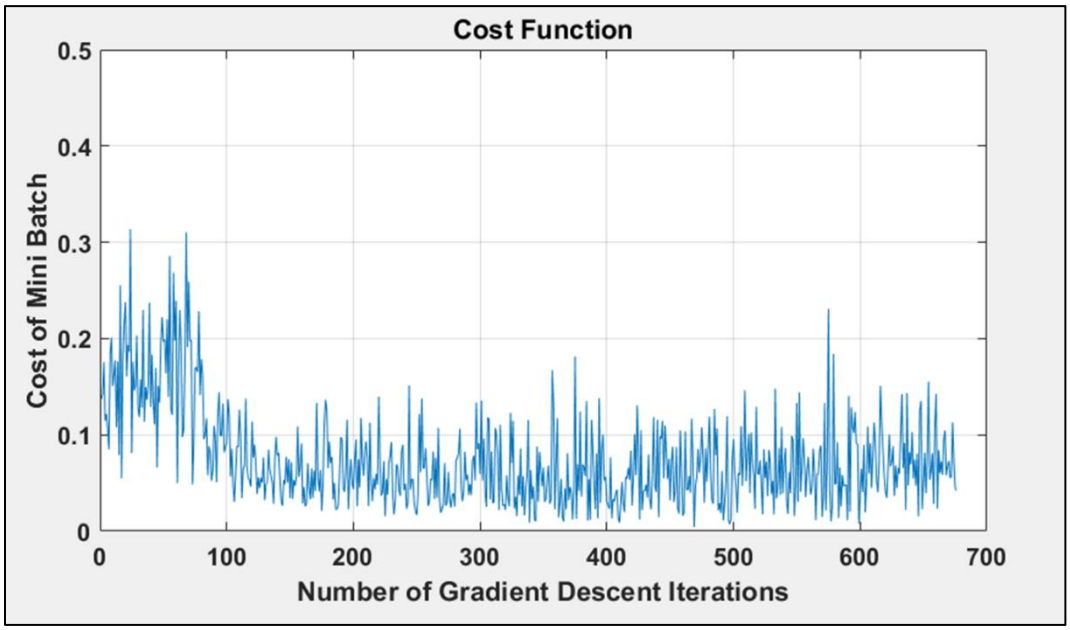


Figure 6.21 - Cost function of the convolutional neural network with a mini batch size of 20, number of minimum episodes per mini batch of 5 and a memory capacity of 1

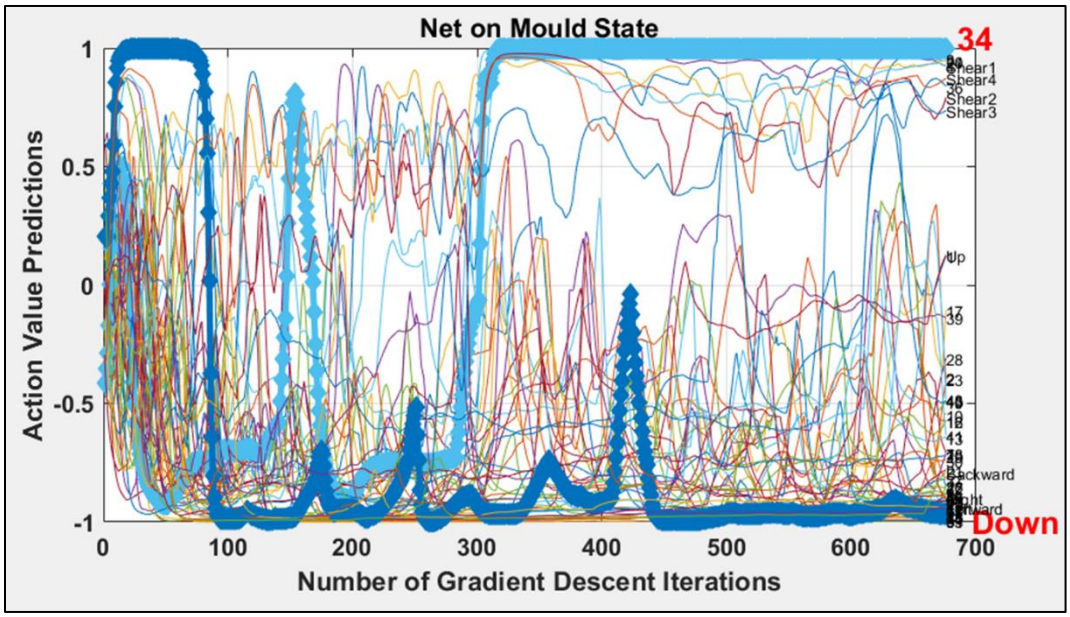


Figure 6.22 - Action value predictions for the 'net on mould state' with a mini batch size of 20, number of minimum episodes per mini batch of 5 and a memory capacity of 1

6.4.4 Effect of the Camera Frame Size on the Simulation

The camera frame size is the number of pixels in each side of each of the three views describing a state. Larger the camera frame size, higher the total number of input

parameters to the CNN. Therefore, higher the computational cost of passing the input states through the CNN. In order to minimize the computational cost, the camera frame size must be kept as low as possible. However, there is a limit to how small the camera frame size can be. This is dictated by the diameter of the nodes making up the pin jointed net. The camera frame size should be large enough such that there is a sufficient number of pixels in each view making up a state to accurately capture the mesh positions of all nodes. Larger node diameters allow the camera frame size to be lowered, and thereby decrease the computational cost. However, having node diameters that are very large can make the pin jointed net behave in an unnatural manner, since adjacent nodes tend to come in to contact with each other during out of plane bending and shear at very low bending and shear angles which limits the bending and shear deformations of the pin jointed net.

6.4.5 Effect of the Weight Swap Steps (C) on the Simulation

This hyper-parameter was introduced in to the simulation as it has been suggested in [116] that it is crucial to tackle the instability that occurs in reinforcement learning when a non-linear function approximator such as a neural network is used to represent the action-value. In [116], C has been set to 10000, which is significantly larger compared to C=300, which has been used in the simulation presented in this chapter. However, any noticeable instability or divergence was not observed by using such a small vale for C. This observation led to a further investigation to check if the value of C can be set to 1 without any instability in the simulation. The simulation was run with C=1 and all other hyper-parameters set to those stated in Table 5.1. Results are illustrated in Figure 6.23 - Figure 6.26. Figure 6.23 shows that the cost gradually decreases and settles down at a negligible value with fewer oscillations compared to when C was set to 300. Figure 6.24 , Figure 6.25 and Figure 6.26 confirm that the agent has developed an identical understanding of the behaviour of the environment to that of when C was set to 300 with the action ‘Down’ being ranked at the top for the Free Net State and at the bottom for the Net on Mould State and Net on Mould Bent state, and also ‘Action 34’ being ranked at the top for all three of these states. This means that C can be set to 1 with no

instability and final matured results are very much identical to that of when the simulation was run with C=300.

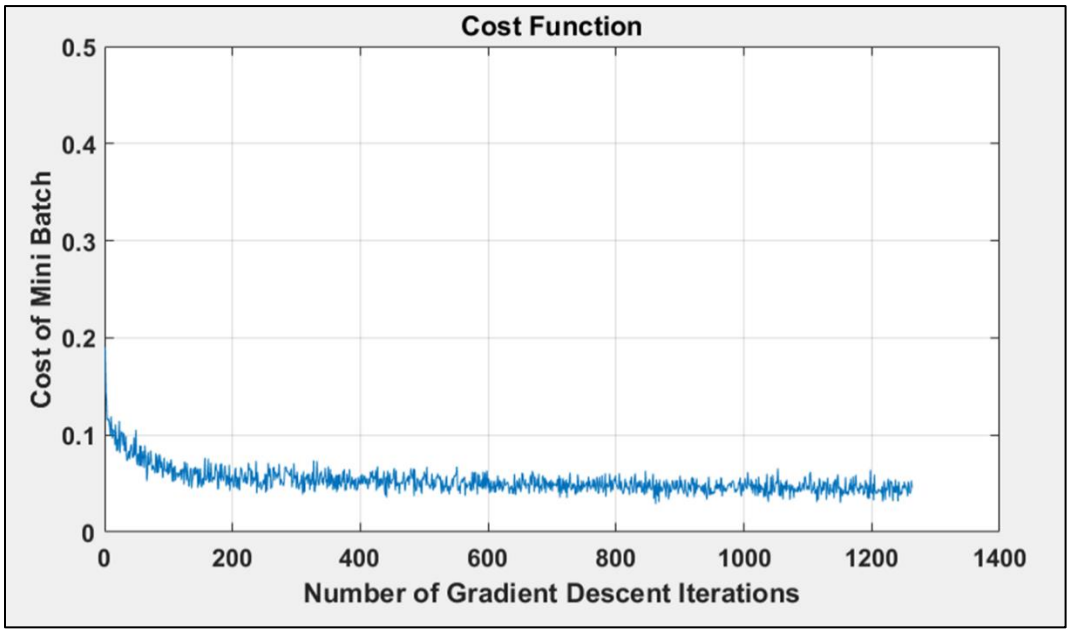


Figure 6.23 – Cost function of the convolutional neural network with weight swap steps (C) = 1

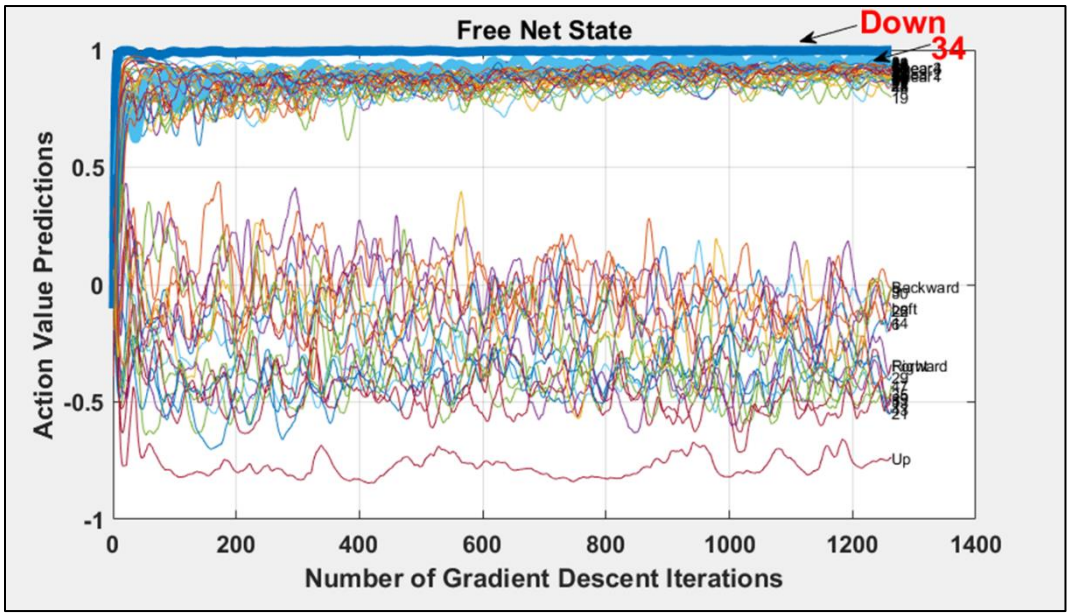


Figure 6.24 - Action value predictions for the 'free net state' with C=1

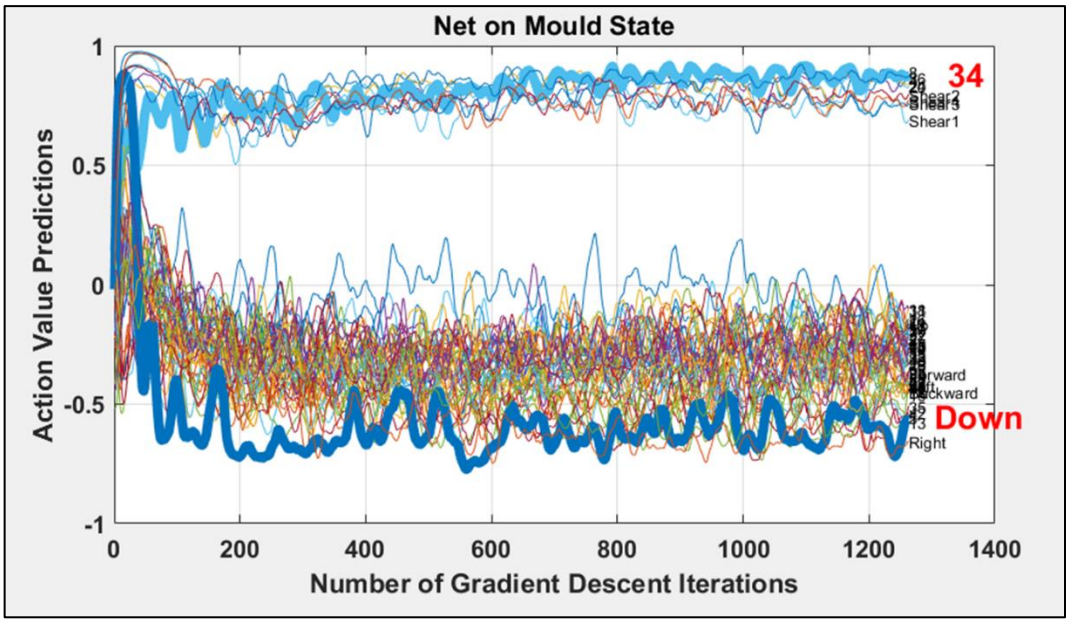


Figure 6.25 - Action value predictions for the 'net on mould state' with C=1

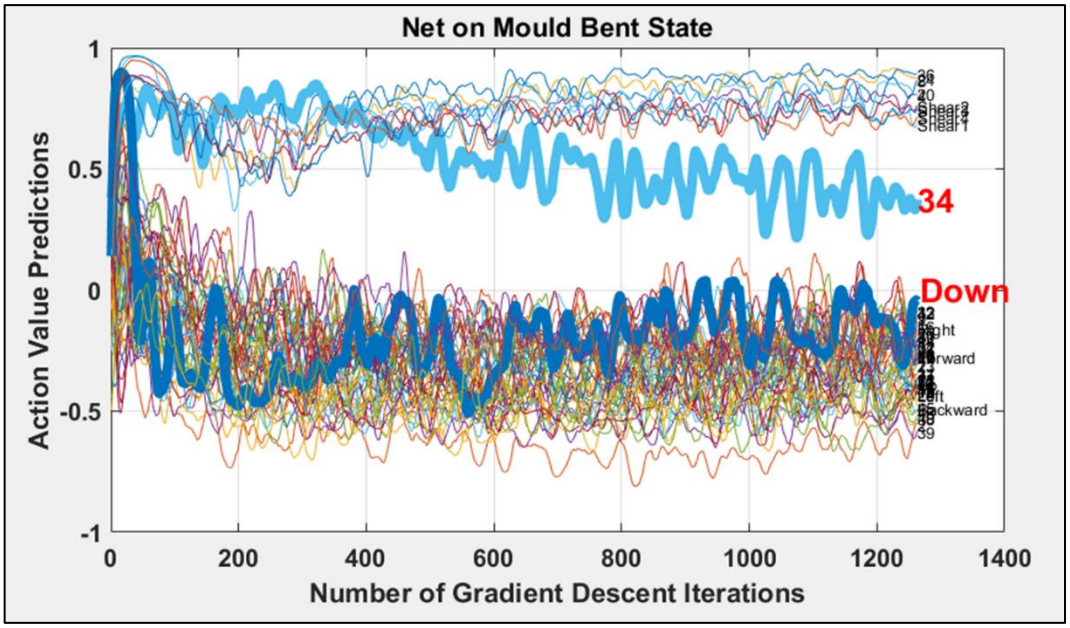


Figure 6.26 - Action value predictions for the 'net on mould bent state' with C=1

6.4.6 Effect of the Rotational Constant, the Translational Step and the Shear Deformation Step on the Simulation

It is crucial to pick the right values for both the rotational constant and the translational step to ensure a successful simulation. Consider solution path 1 explained in section 6.2.1. If the translational step were set too high, the net would submerge in the mould

upon taking one 'Down' action. In this case, the agent will never be able to find a solution. On the other hand, if the translational step was set too low, it would take an unnecessary number of 'Down' actions for the agent to place the net on the mould. The same concept applies for the rotational constant when the agent tries to bend the net to gain contact with the ramp section of the mould. No in plane shear deformation was required to complete this particular layup. However, when it comes to geometries that require shearing of the net, a similar explanation applies to the shear deformation step as well. While it was very straightforward to pick ideal values for both the rotational constant and the translational step for this simple mould shape with a known ideal solution path, this is certainly not a trivial task for more complex geometries. This is certainly one major drawback of this simulation. The obvious solution to this problem is to set both these hyper-parameters to very low values to ensure that the agent does find a solution path. However, this means that the simulation must now be run at a very high mini batch size, which is computationally expensive and time consuming.

6.4.7 Effect of Number of Mini Batch Gradient Descents to Maximum E Greedy on the Simulation

Care must be taken to ensure that this hyper-parameter is not set to a value that is too small. This hyper-parameter determines when the agent starts to use the SOFTMAX action selection criteria. It is good practice to take actions randomly at the beginning and develop a rough understanding of the behaviour of the reinforcement-learning environment before using the SOFTMAX action selection criteria. Consider the example discussed in section 6.3. It is clear that after about 40 learning iterations, the agent has developed a good understanding of the behaviour of the environment, since grouping of actions according to positive/negative rewards has begun. Therefore, it makes sense to deploy the SOFTMAX action selection criteria beyond this point. This is the reason why this hyper-parameter was set to 75 in this example. If this hyper-parameter was set to a very low value, say 5, the agent might not explore actions that are initially ranked low due to random initialization of weights of the CNN, but actually does yield high future cumulative rewards. At the beginning of a fresh simulation, it is safe to set this

hyper-parameter to a relatively large value, then observe results of the simulation while the agent takes actions in a completely random manner, wait for visible structure in the results and then rerun the simulation with an appropriate value for this hyper-parameter.

6.4.8 Effect of the Memory Capacity on the Simulation

Size of the memory capacity must always be set relative to the number of minimum episodes per mini batch. It can be recommended that the memory capacity should not be larger than 30% of the size of the number of minimum episodes per mini batch. If the memory capacity is set too high, majority of a mini batch will be made up of experience from the agent's long-term memory. This leaves little room for the agent to explore new solution paths. In addition, gradient descent steps will be biased towards sequences of actions contained in the agent's memory. This will lead to a failure in the agent's ability to find a more general and global minimum to the cost of the CNN.

6.5 Summary

Results presented in this chapter suggest that the proposed artificially intelligent layup agent has successfully determined the ideal layup sequence for the mould shape it was presented with. In addition, through a convolutional neural network, it was capable of building up an accurate representation of the reinforcement-learning environment, in the form of a mapping between states and action values. However, it was later explained that a multitude of hyper-parameters had to be fine-tuned to ensure the success of the AI agent. Furthermore, many of the hyper-parameter values required to ensure the success of the AI agent are dependent on the mould shape and net resolution. This could be a major drawback in ensuring the success of the AI agent, when moulds with more complex geometries are in consideration. For example, failing to set hyper-parameters such as Rotational Constant, Translational Step, and Shear Deformation Step to ideal values for a given mould shape may result in the non-existence of a layup solution or time to achieve a solution drifting towards an unmanageable amount. This issue is explored in the next chapter in more detail.

The issue with determining ideal values for hyper-parameters, Rotational Constant and Translational Step may be solved by restricting node submersion in the mould. Achieving this would require a slightly different model for the interaction between the net and the mould as opposed to the model presented in this thesis. In such a model, the mould and the nodes making up the net would have solid surfaces and there would not exist any mechanism by which nodes could submerge in the mould. The exploration of such a model is suggested as future work.

It was also made clear that a certain element of 'luck' through the SOFTMAX action selection criteria is required for the AI agent to come across the ideal action sequence. Since, only three actions were required to layup the mould shape used in this chapter, and only 58 actions were available to the agent due to the low resolution of the net used, given enough time, the agent is bound to find the ideal layup sequence, especially with the SOFTMAX action selection criteria in place. However, the same case may very well not apply when a more complex mould shape that requires a larger number of actions to layup is in consideration. Nevertheless, the agent, through the simulation will always develop an accurate representation of the environment, in the form of a mapping between states and action values, which will give an idea of which set of actions is most suitable for a given state. This knowledge maybe valuable in determining the ideal action sequence with the aid of a human laminator.

Since the agent successfully learned the sequence of actions required to layup this simple ramp shape, it can be argued that the agent can also successfully learn the sequence of actions required to layup other singly curved geometries such as a cylinder (draping of only the curved surface), since only a sequence of out of plane straight bending actions are required to cover the curved surface of a cylinder. However, the same argument cannot be made for a singly curved geometry such as a cone, since covering the curved surface while ensuring a near net shape drape would require the initial shape of the pin jointed net to be not a square, or the net to be trimmed after the completion of the drape. This is not a capability that is available in the current model. Adding such capabilities is suggested as future work.

CHAPTER 7: An Artificially Intelligent Layup Agent: Advanced Case Studies

7.1 Introduction

In this chapter, the author attempts to further explore working mechanics and limitations of the proposed artificially intelligent layup agent using two distinct case studies: a spherical mould and the 3D mould used for VR training in Chapter 4 of this thesis. For both case studies, the approach taken in modelling the reinforcement learning environment and the thought process behind choosing hyper-parameters will be discussed. Action value predictions from relevant reinforcement-learning simulations will be used to prove the ability of the AI agent to develop a representation of the behaviour of the environment. Finally, layup sequences for both mould shapes, as provided by the AI agent will be analysed to understand drape limitations of the simulation in its current state.

7.2 Case Study 1 – Spherical Mould

7.2.1 Modelling the Reinforcement-Learning Environment

A spherical mould in particular has been chosen as the next case study, since it is the most studied geometry in the literature [147], in terms of draping woven fabric, and hence would be a good reference point. In addition, draping a spherical mould without wrinkling the pin jointed net would require in-plane shear, and hence, this will be a good case study to explore the drape capabilities and limitations of the current pin jointed net model. A spherical mould with a diameter of 1 unit was used. The reinforcement was modelled as a rigid pin-jointed net with resolution 5x5 (that is 25 nodes in total). Each spherical node has a diameter of 0.08 units and each bar connecting two nodes has a length of 0.25 units. These dimensions are illustrated in Figure 7.1.

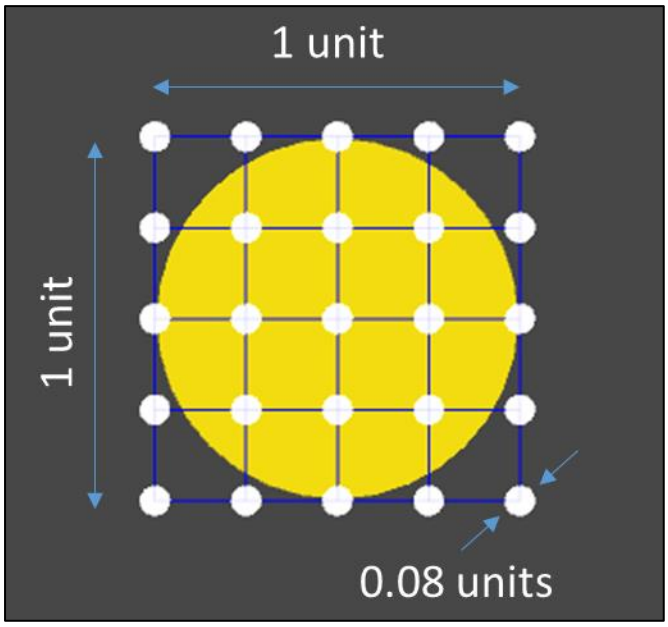


Figure 7.1 - Dimensions of the mould and the pin jointed net (top view)

A sphere collider with a diameter of 1 unit is attached to the mould to detect contact between the mould and the net. A dummy collider with a dimeter of 0.92 is also attached to the mould to detect node submersions as explained in section 5.2.1. These dimensions are illustrated in Figure 7.2. A box collider with dimensions of 1.8 x 1.8 x 1.8 units is also used to ensure that a finite reinforcement-learning environment is used similar to that explained in the preliminary case study. This is illustrated in Figure 7.3. Three cameras are placed on the top and to the right and left of the mould to capture the three views for state representations.

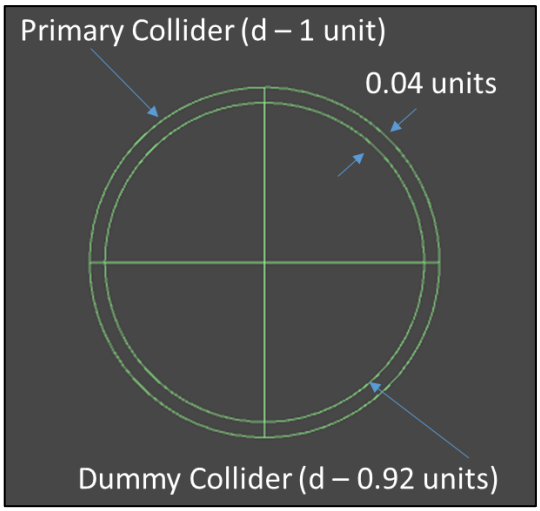


Figure 7.2 - Top view of colliders attached to the mould and their dimensions

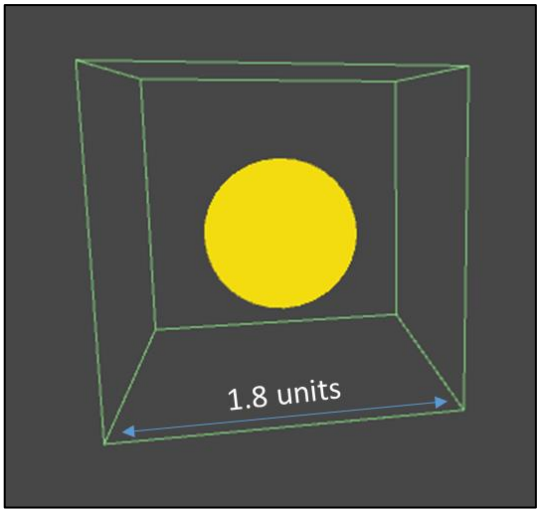


Figure 7.3 - Box collider used to create a finite reinforcement-learning environment

7.2.2 Hyper-Parameters Used in the Simulation

Hyper-parameters used for this case study are illustrated in Table 7.1. It is worth comparing these hyper-parameters against those used in the preliminary case study to understand how they are affected by the mould shape and the net resolution.

The Locking Angle has been reduced from 30° to 15°. A lower locking angle is required to ensure sufficient shear deformation can be applied to the net in order to bring all nodes in to contact with the mould surface. This will be explained in detail in section 7.2.5.

The Rotational Constant has been reduced from 22.5^o to 20^o. It was noticed that with a Rotational Constant of 22.5^o, a solution was not achievable due to node submersion. Through trial and error, an optimal value of 20^o was chosen.

No change has been made to the Camera Frame Size. This is because, node diameter has been kept constant (0.08 units) between the two simulations. Refer to section 6.4.4 for further clarification.

No change has been made to the Maximum E Greedy Value. This parameter is always set to 1, since it is never influenced by the mould shape or the net resolution. An initial trial run with the Number of Mini Batch Gradient Descents to Maximum E Greedy set to a very large value (>1000) indicated that the agent was capable of learning an accurate mapping between states and action values in less than 50 gradient descent iterations. This will be further explained in the results section of this case study. Hence, this hyper-parameter was set to 75 (unchanged from the preliminary case study).

Mini Batch Size has been increased from 300 to 600. Number of Minimum Episodes per Mini Batch has been kept unchanged at 30, resulting in the Maximum Number of Actions per Episode to increase from 10 to 20. In the preliminary case study, it was known that only three actions were required to complete the layup. However, in this case study, this value was estimated to be between 10 and 15 through initial trials. Due to this reason, the Maximum Number of Actions per Episode had to be set at 20. Therefore, in keeping the Number of Minimum Episodes per Mini Batch unchanged from the preliminary case study, the Mini Batch Size had to be increased to 600.

No change has been made to the Shear Deformation Step. It was noticed from the results of the simulation on the spherical mould that no shear deformation has been exploited by the AI agent. However, the final score of the simulation can be maximized by applying in plane shear on to the PJN. This will be further explored in a subsequent section of this chapter.

The translational step has been set to 0.25. The initial state of the environment (Free Net State) has been designed such that performing a single 'Down' action will ensure contact between the mid node of the net and the mould. This is illustrated in Figure 7.4.

The Memory Capacity has been increased from three to five. Due to the increase in complexity of the mould shape from the preliminary case study to the current case study, it was expected that the AI agent would be able to find multiple layup solution paths. Due to this reason, the agent has been given a higher Memory Capacity.

No change has been made to the Discount Rate. This parameter is always set to 0.99, since it is never influenced by the mould shape or the net resolution.

Learning Rate has been increased from 0.01 to 0.06. Due to the increase in the resolution of the net from 3x3 to 5x5, the total number of actions available to the agent has increased from 58 to 106. This causes the architecture of the CNN used for the AI agent to change such that the number of neurons in the output layer increases from 58 to 106. It was determined through trial and error that the ideal learning rate for this particular CNN architecture is 0.06.

Weight Swap Steps was set to one, since it was concluded in the hyper-parameter study in section 6.4 that it is unnecessary to set this hyper-parameter to any other value.

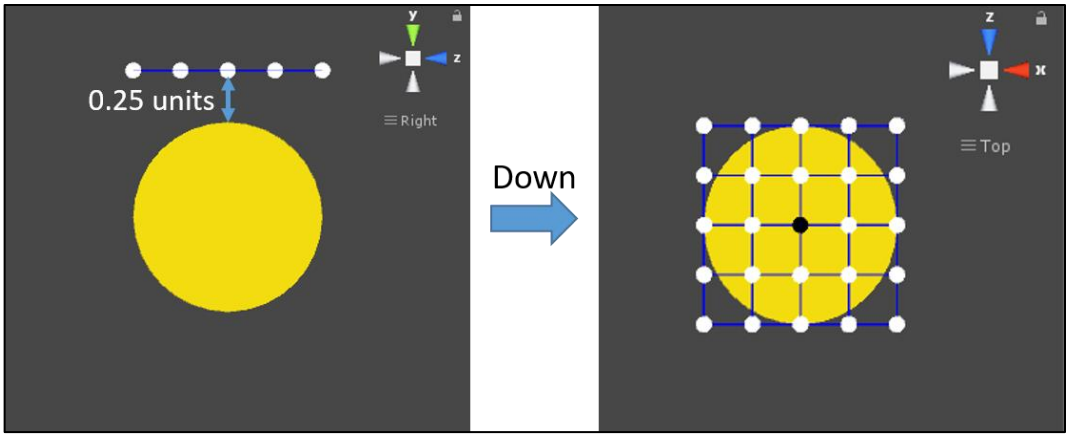


Figure 7.4 – Illustration of the initial state of the environment (left) and the state of the net after performing one 'Down' action (right)

Table 7.1 - Hyper-parameters used for case study 1 - spherical mould

Hyper-Parameter Name	Value	Hyper-Parameter Name	Value
Locking Angle	15°	Maximum Number of Actions per Episode	20
Rotational Constant	20°	Shear Deformation Step	0.75
Camera Frame Size	84	Translational Step	0.25
Maximum E Greedy Value	1	Memory Capacity	5
Number of Mini Batch Gradient Descents to Maximum E Greedy	75	Discount Rate (ϒ)	0.99
Mini Batch Size	600	Learning Rate (α)	0.06
Number of Minimum Episodes per Mini Batch	30	Weight Swap Steps (C)	1

7.2.3 Proof of the Reinforcement Learning Algorithm

The simulation was left to run overnight up to 1400 gradient descent iterations. Action value predictions for the Free Net State and the Net on Mould State as illustrated in Figure 7.5 and the cost function were plotted against the number of gradient descent iterations.

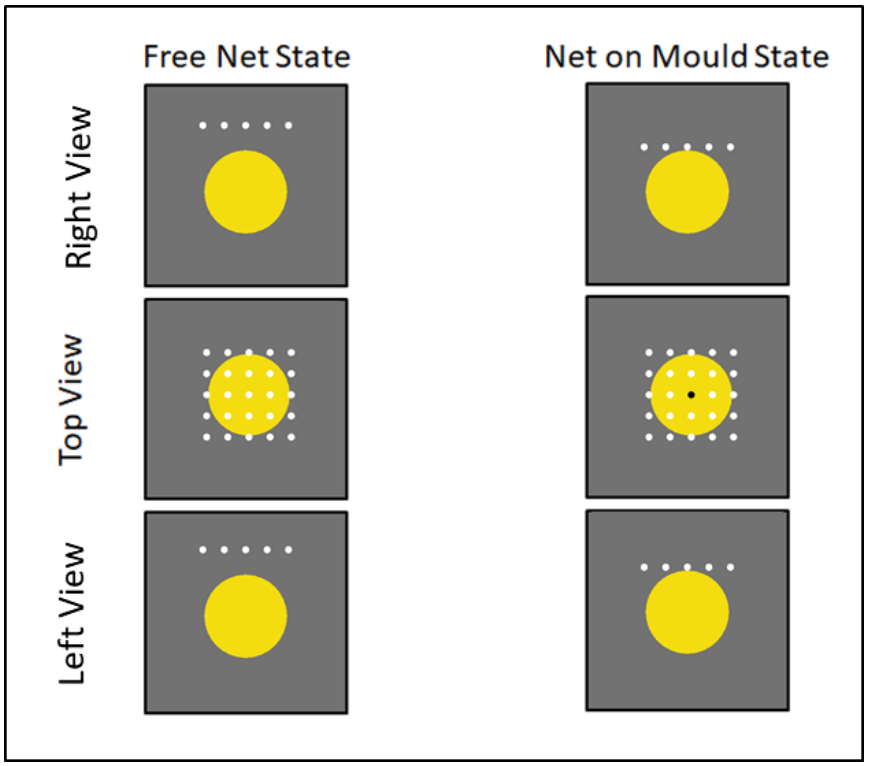


Figure 7.5 - State representations for free net and net on mould states using right, top and left views

For the simulation to be successful, the agent must be capable of distinguishing between the Free Net State and the Net on Mould State. A simple way to check this is to analyse the action value predictions for the six translational actions of these two states. In a successful simulation, action value predictions for the translational actions of the Net on Mould State must be lower than that of the Free Net State, since all translational actions result in a negative reward when performed from the Net on Mould State.

Action value predictions for these two states from the simulation are illustrated in Figure 7.6 and Figure 7.7. In these two figures, translational actions are highlighted in red colour for clarity. For the Free Net State, translational actions are ranked higher up in the region of action value predictions of 0.6-1. In contrast, for the Net on Mould State, translational actions are ranked relatively lower down in the region of action value predictions of (-0.2)-0.4. Thus, it can be confirmed that the agent has developed an accurate representation of the behaviour of the environment.

Another indication of a successful simulation is the gradual decrease and stabilization of the cost function at a negligible value as previously discussed in the preliminary case study. A similar pattern in the cost function was observed in this simulation and this is illustrated in Figure 7.8. This also confirms that a suitable learning rate has been picked for the simulation.

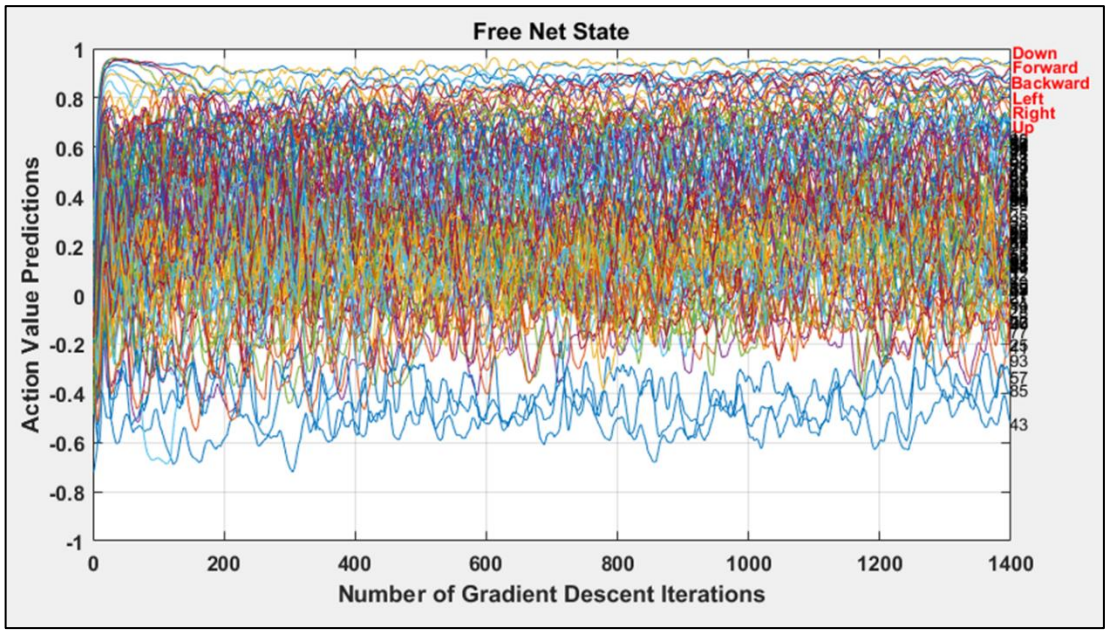


Figure 7.6 - Action value predictions for the 'free net state' up to 1400 gradient descent iterations

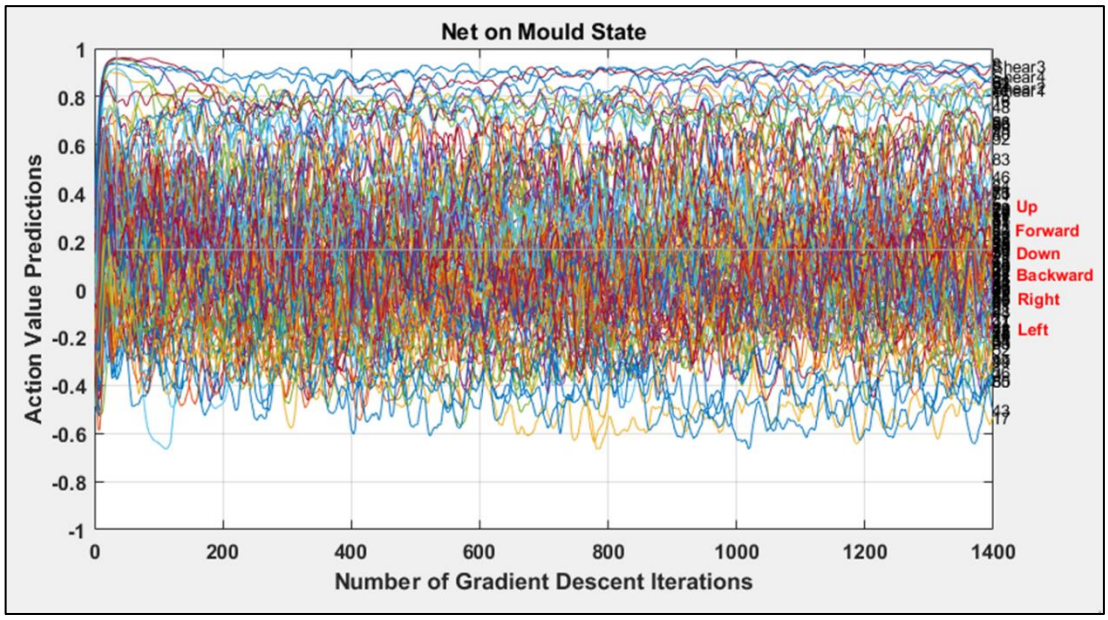


Figure 7.7 - Action value predictions for the 'net on mould state' up to 1400 gradient descent iterations

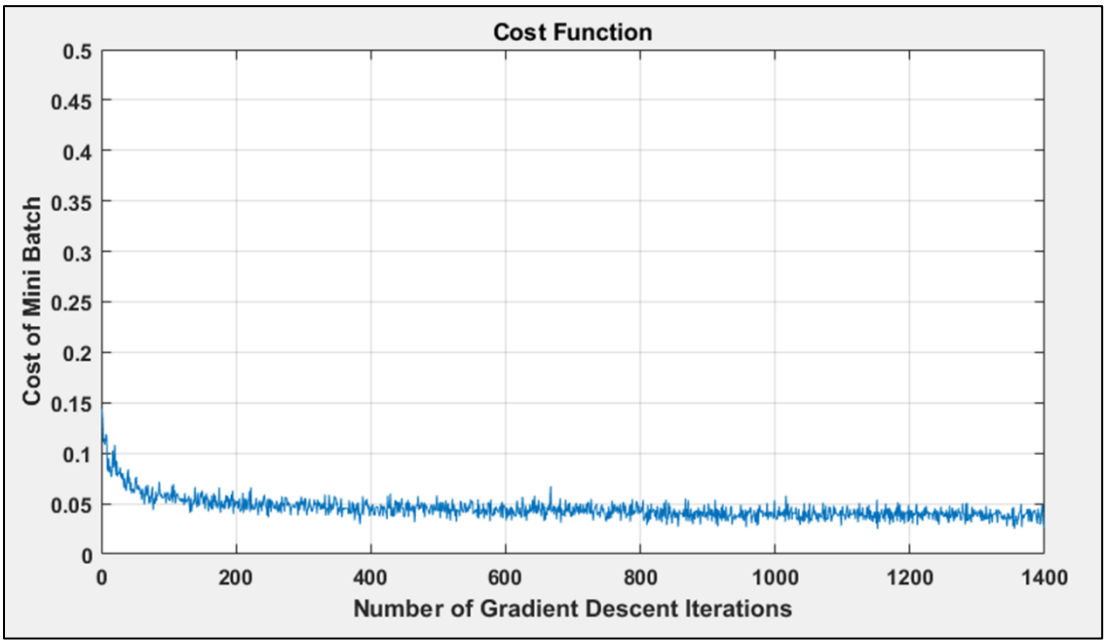


Figure 7.8 - Cost function of the convolutional neural network with a learning rate of 0.06

7.2.4 Analysing Layup Solution Paths

Since the Memory Capacity was set at five at the beginning of the simulation, five optimized layup sequences are output at the end of the simulation. End states of these five solutions as obtained from the simulation are illustrated in Figure 7.9. Numerical output statistics from the simulation on these solutions and comments are illustrated in Table 7.2. It should be noted that the same exact solutions could not be reproduced with such a simulation, since the AI agent always takes actions in a random manner. If the simulation was run again, the agent is very likely to discover five completely different solution paths. However, if the memory capacity was set to a large enough value, it may be possible to obtain and reproduce all possible solution paths. A larger memory capacity means that the mini batch size will also have to be increased. This is associated with an increase in the computational cost to run the simulation.

It is worth restating the primary objective of the simulation, which is to maximize the cumulative reward the agent receives using the least number of actions. Therefore, with this objective in mind, it can be concluded that solution path 5 is the preferred layup sequence for this spherical mould. Both solution paths 5 and 1 have the same cumulative

reward of 0.52. However, solution path 5 is achieved using seven actions, which is twelve actions less than what is used in solution path 1. A simple visual inspection of the two solutions also suggest that solution path 5 should be preferred to solution path 1, since it is a symmetrical layup pattern.

While solution path 5 seems desirable compared to the rest of the solutions, 12 out of 25 nodes have failed to make contact with the mould in solution path 5. In its current state, the remaining nodes of the net cannot come in to contact with the mould since any further out of plane diagonal bending is restricted by wrinkling induced by bending-bending coupling. This issue is discussed in detail in the next section.

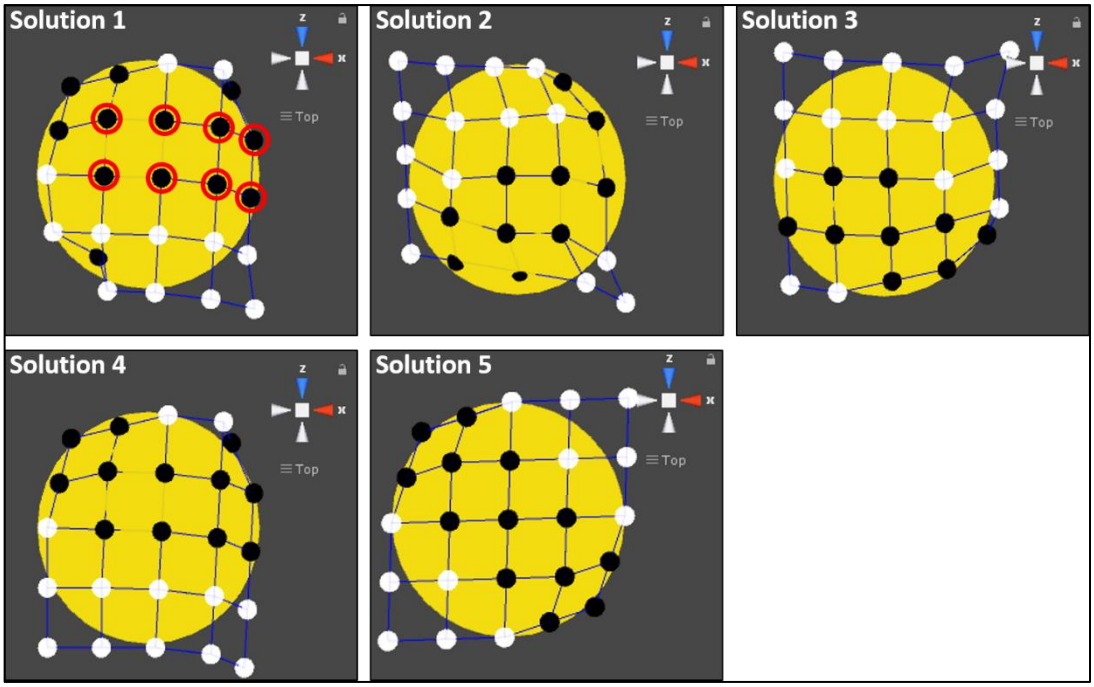


Figure 7.9 - End states (top views) of the five solutions obtained from the simulation

Table 7.2 – Numerical output statistics and comments on simulation results

Solution No.	No. of Actions	Cumulative Reward	Comments
1	19	0.52	Initially, the net has been draped over the mould along the fibre direction to ensure contact between the eight nodes circled in red in Figure 7.9. Rest of the nodes have gained contact through out of plane diagonal bending.
2	20	0.4	The net has been draped over the mould along a diagonal direction. In plane shear has been applied along the opposite diagonal direction, which has caused some nodes to lose contact with the mould.
3	11	0.36	This must be an incomplete solution since not many nodes have gained contact compared to other solutions. If the simulation was allowed to run for a longer period, the agent would have succeeded in ensuring that more nodes gained contact with the mould. The net has been draped over the mould in the fibre direction.
4	19	0.48	This solution is very similar to solution 1. The net has been draped over the mould along the fibre direction. Solution 4 would converge to solution 1 if the simulation was left to run for a longer period.
5	7	0.52	The net has been draped over the mould along the opposite diagonal direction to that used in solution 2. This is a symmetric layup pattern.

7.2.5 Understanding Drape Limitations

It was highlighted in the previous section that for solution path 5, 12 out of 25 nodes have failed to make contact with the mould. In this section, the sequence of actions required to bring all 25 nodes in to contact with the mould will be presented and it will be shown why the AI agent will fail to obtain this result with the current combination of hyper-parameters used.

The surface to be laid up on a spherical mould is doubly curved. Therefore, to ensure that the entire net is draped over this surface without wrinkles, certain amount of in plane shear deformation is required. Therefore, the net is first sheared until the locking angle is reached. This is illustrated in Figure 7.10. The next step would be to translate the net downwards to secure it to the mould. It was mentioned in section 7.2.2 that the Translational Step was intentionally set at 0.25 and the net was placed 0.25 units above the mould surface to ensure that a single 'Down' action would ensure contact between the mid node of the net and the top of the mould surface. This is no longer possible, since shearing of the net has moved the mid node away from the centre of the mould. It is important that the mid node gains contact with the top of the mould to ensure that a symmetric layup is achieved, especially in the case of laying up a hemisphere. A single 'Down' action performed from this sheared state results in no nodes making contact with the mould, since there are now no nodes at the top of the mould surface. Two 'Down' actions result in the submergence of the net in the mould as illustrated in Figure 7.11. This will only lead to the termination of that particular learning episode.

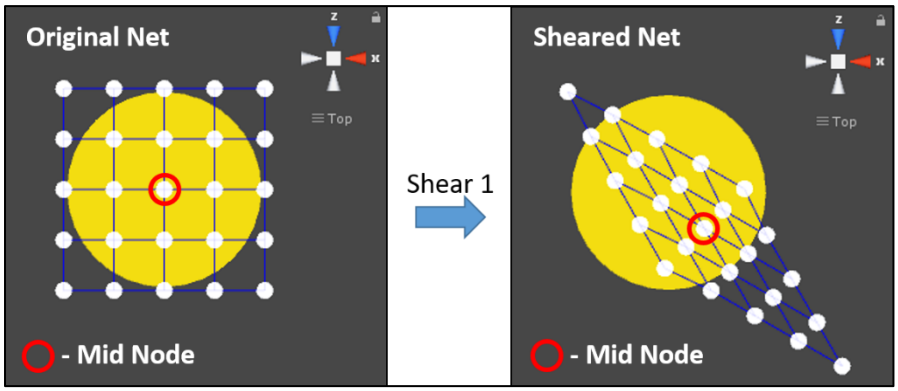


Figure 7.10 – States of the net (top view only) before and after applying in plane shear

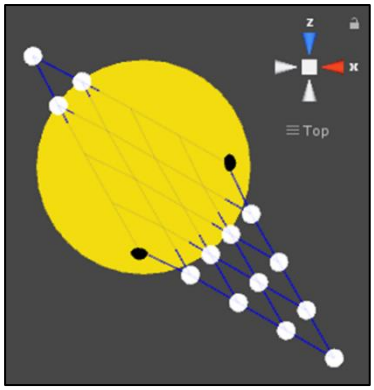


Figure 7.11 - Submergence of the net (top view) in the mould after two consecutive 'Down' actions

The agent can attempt to re-centre the sheared net by performing one 'Left' action and one 'Up' action. However, with the combination of hyper-parameters used in this simulation, a perfect re-centring is not possible as illustrated in Figure 7.12.

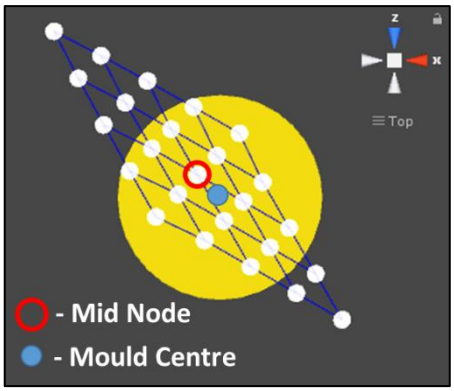


Figure 7.12 - Re-centred sheared net

Given that the agent was able to re-centre the net perfectly and translate it down to ensure contact between the mid node and top of the mould surface, next step would be to perform appropriate diagonal out of plane bending actions to drape the sheared net on to the mould surface. Final draped pattern is illustrated in Figure 7.13. Note that 2 nodes still fail to make contact with the mould due to wrinkling caused by bending-bending coupling. In performing diagonal out of plane bending actions, nodes are rotated about pivots placed on diagonal nodes and mid points between diagonal nodes as explained in section 5.2.1. In a sheared net, moment arms for diagonal bending in the sheared direction are larger compared to a net with no shear due to the elongation of the net in the sheared direction. This means that upon each diagonal bending action, nodes will move a larger distance in the 'y' direction. This is the reason for node submergence observed in Figure 7.13. Since the learning episode is terminated upon node submergence, the agent will fail to find the action sequence required to achieve this final drape pattern.

Both issues related to translation of the net and out of plane diagonal bending can be solved by using smaller values for the Translational Constant and the Rotational Constant. But, it is certainly not intuitive what these values must be without knowledge of the ideal drape pattern. As previously mentioned in section 6.4.6, setting these hyper-parameters to small values present issues related to computational resources and simulation time. Similar issues arise when setting the Shear Deformation Step.

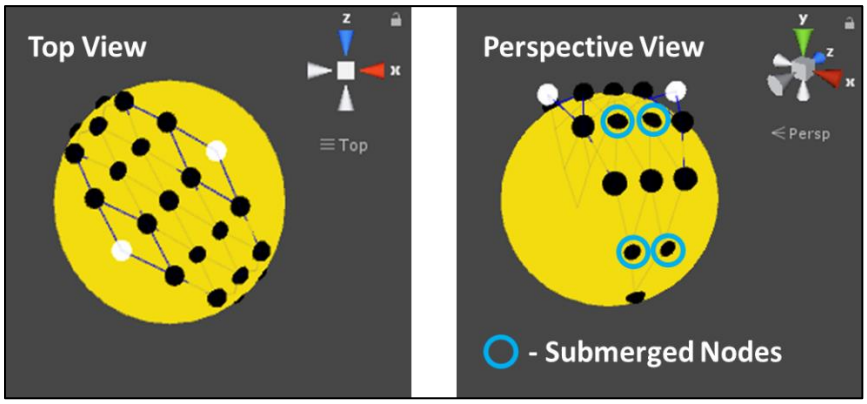


Figure 7.13 - Final drape pattern of the sheared net

7.3 Case Study 2 – 3D Mould used for VR Training

In this case study, the same 3D mould used for VR training is used. This mould shape was chosen for the second case study, since it would produce insights on the usability of the layup instructions learnt by the AI agent in creating VR training aids. Dimensions of this mould were provided in section 4.2.1.

7.3.1 Modelling the Reinforcement-Learning Environment

The mould was re-scaled to have length and width of 1 unit. The reinforcement was modelled as a rigid pin-jointed net with resolution 5x5 (that is 25 nodes in total). Each spherical node has a diameter of 0.08 units and each bar connecting two nodes has a length of 0.25 units. These dimensions are illustrated in Figure 7.14.

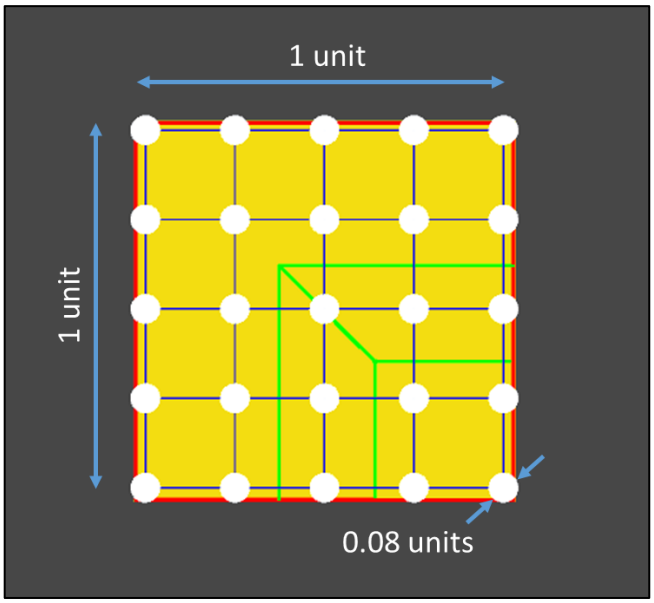


Figure 7.14 - Dimensions of the mould and the pin jointed net (top view)

A mesh collider is attached to the mould to detect contact between the mould and the net. A dummy collider with an offset of 0.04 from the original collider is also attached to the mould to detect node submersions as explained in section 6.2.1. A box collider with dimensions of 1.8 x 1.8 x 1.8 units is also used to ensure that a finite reinforcement-learning environment is used similar to Case Study 1. Four box colliders are placed to detect contact between nodes and the sides of the mould as illustrated in Figure 7.15. Three cameras are placed on the top, side and front of the mould to capture the three views for state representations.

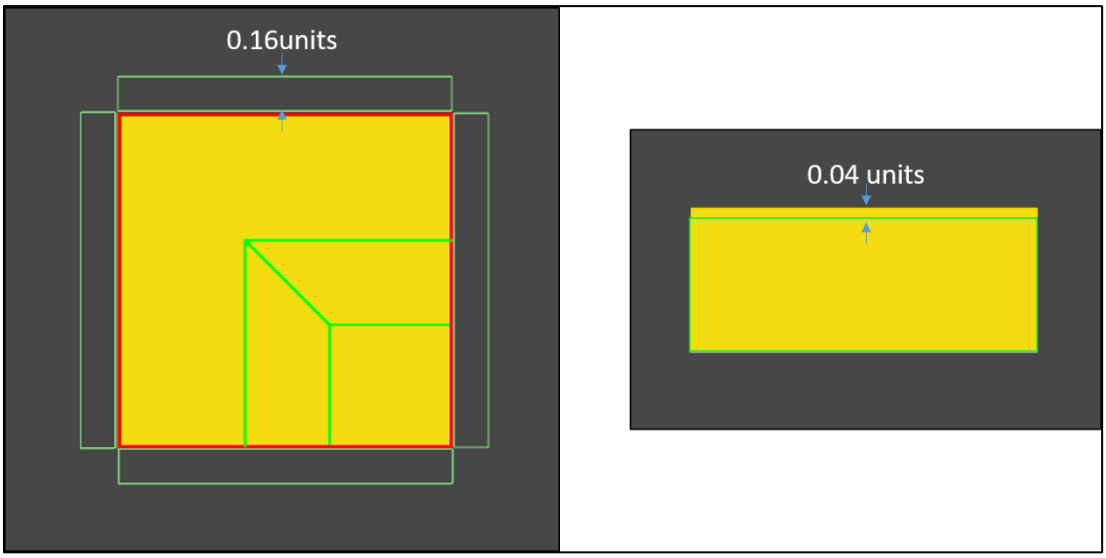


Figure 7.15 - Positioning of box colliders (left - top view, right – side view) to detect contact between nodes and the sides of the mould

7.3.2 Hyper-Parameters Used in the Simulation

Hyper-parameters used for this case study are illustrated in Table 7.3. Changes in these hyper-parameters in comparison to the first case study are discussed below.

The Rotational Constant has been reduced from 20^{o} to 10^{o}. This was necessary to avoid node submergence that would have caused the non-existence of a layup solution.

It was estimated that the total number of actions required to acquire a layup solution was in the region of 5-10. Therefore, Mini Batch Size was reduced from 600 to 450, keeping the Number of Minimum Episodes per Mini Batch unchanged at 30, which results in a Maximum Number of Actions per Episode of 15.

Memory Capacity was reduced from 5 to 3, since not many distinct solution paths were expected for this particular mould.

Table 7.3 – Hyper-parameters used in the simulation for case study 2

Hyper-Parameter Name	Value	Hyper-Parameter Name	Value
Locking Angle	15°	Maximum Number of Actions per Episode	15
Rotational Constant	10°	Shear Deformation Step	0.75
Camera Frame Size	84	Translational Step	0.25
Maximum E Greedy Value	1	Memory Capacity	3
Number of Mini Batch Gradient Descents to Maximum E Greedy	75	Discount Rate (ϒ)	0.99
Mini Batch Size	450	Learning Rate (α)	0.06
Number of Minimum Episodes per Mini Batch	30	Weight Swap Steps (C)	1

7.3.3 Proof of the Reinforcement-Learning Algorithm

The simulation was left to run up to 500 gradient descent iterations. Action value predictions for the Free Net State and the Net on Mould State as illustrated in Figure 7.16 and the cost function were plotted against the number of gradient descent iterations. It should be noted that in case study 1, the simulation was run up to 1400 gradient descent iterations as opposed to 500 in case study 2. The decision on when to stop the simulation is made based on the following two conditions been met, in the listed order:

1. The cost function has settled down to a negligible value and is no longer decreasing.
2. Expected cumulative reward has been achieved by the AI agent.

It is clear from Figure 7.17 and Figure 7.18 that the AI agent has successfully distinguished between the Free Net State and the Net on Mould State, since all six translational actions are ranked at the top for the Free Net State and vice versa for the Net on Mould State. Figure 7.19 confirms that the cost function has settled down at a negligible value at 500 gradient descent iterations.

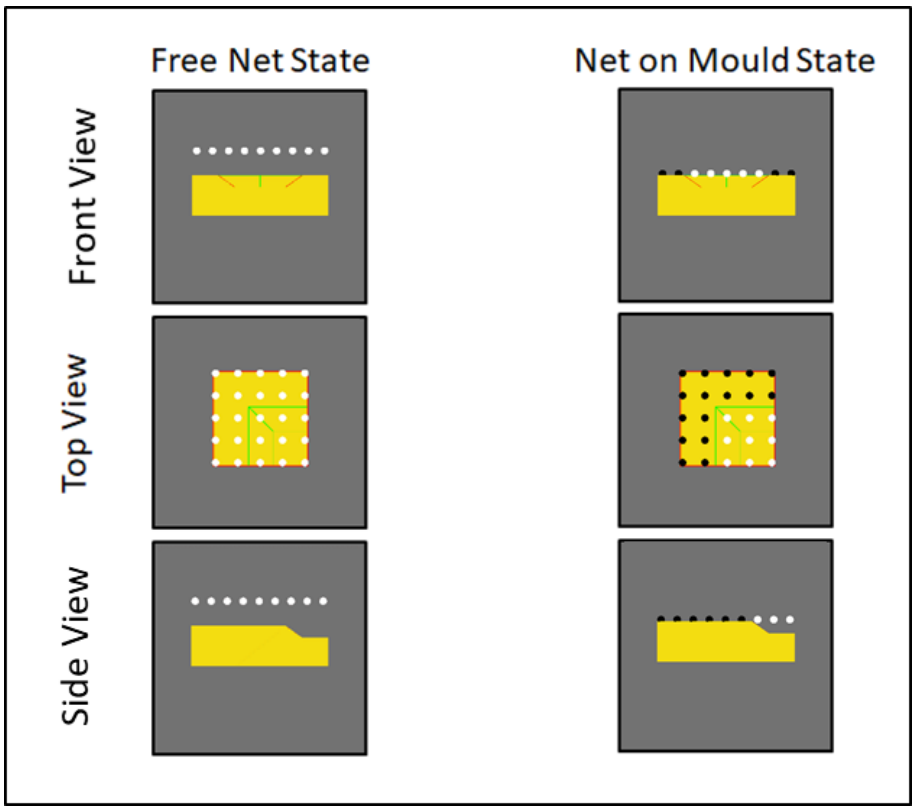


Figure 7.16 - State representations for free net and net on mould states

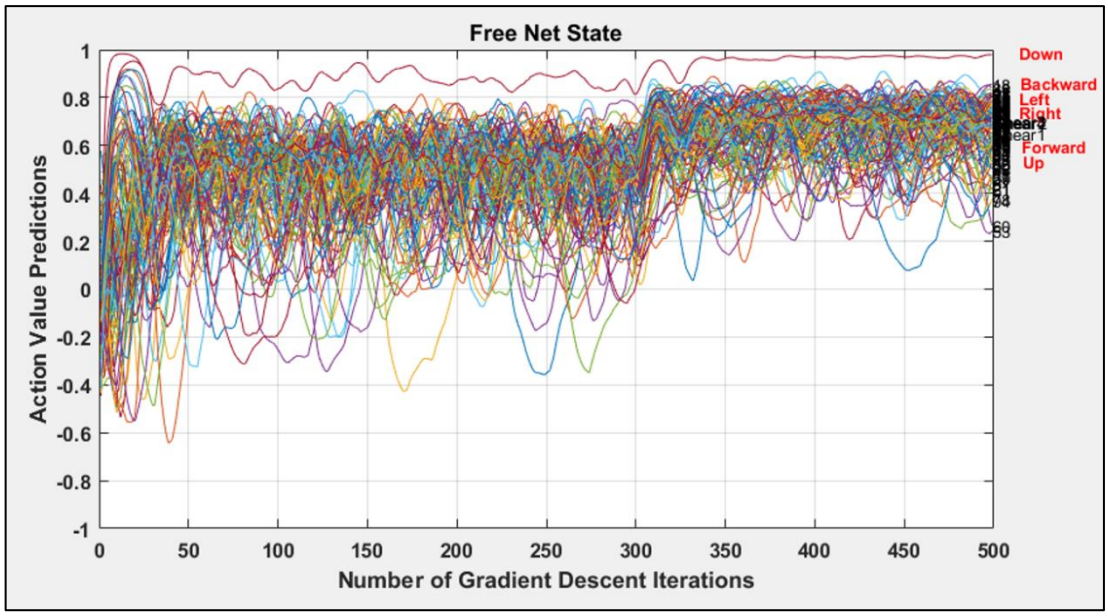


Figure 7.17 - Action value predictions for the 'free net state' up to 500 gradient descent iterations

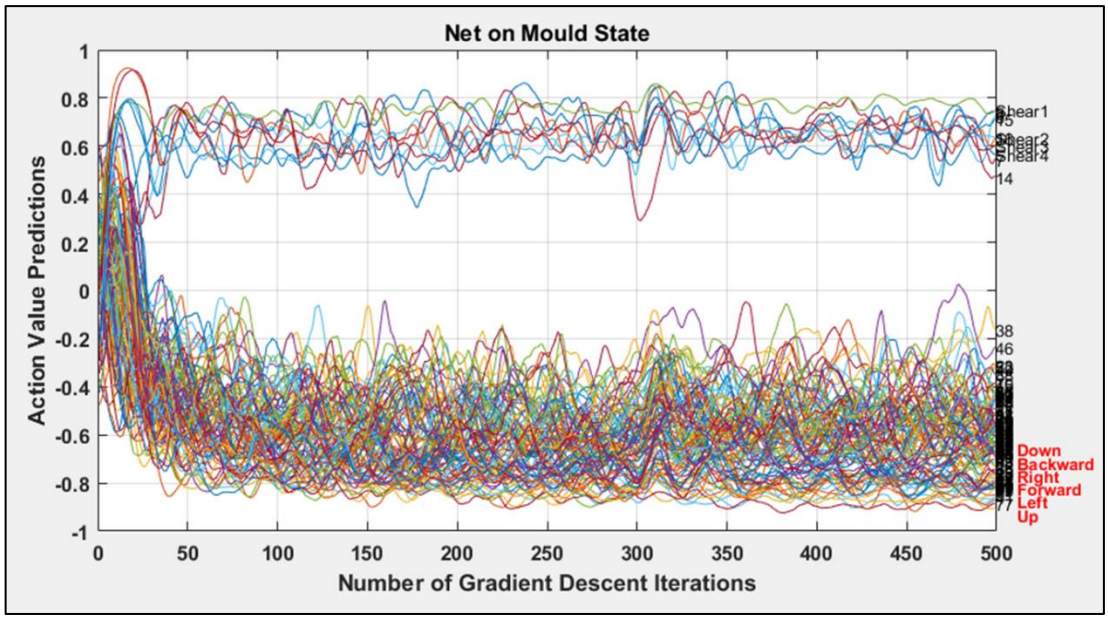


Figure 7.18 - Action value predictions for the 'net on mould state' up to 500 gradient descent iterations

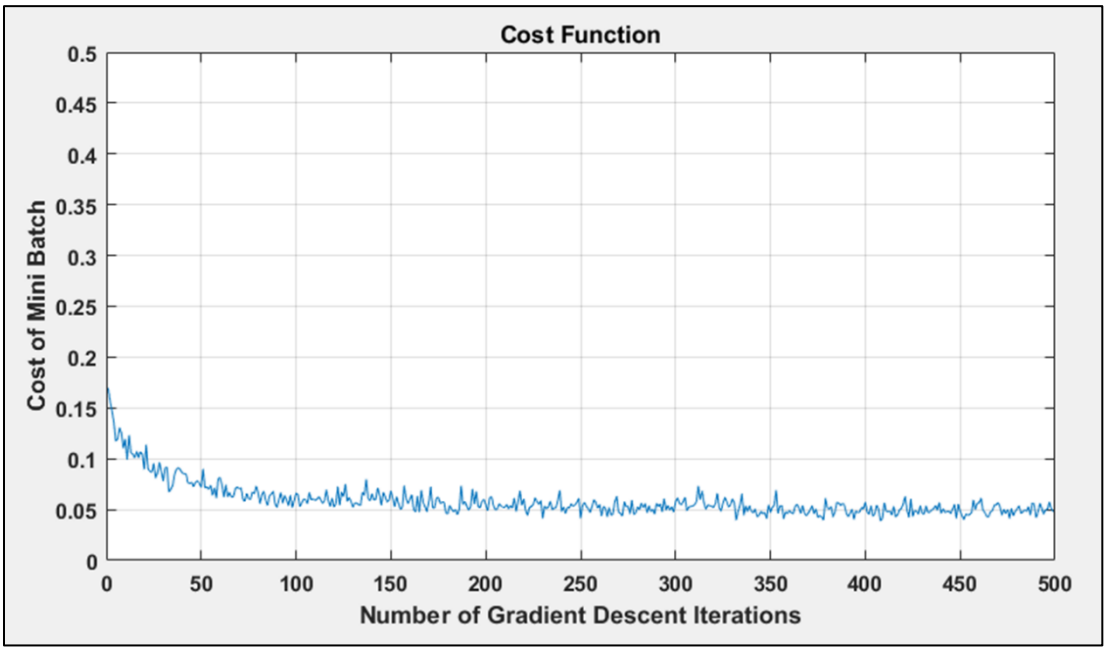


Figure 7.19 - Cost function of the convolutional neural network with a learning rate of 0.06

7.3.4 Analysing Layup Solution Paths

Since the Memory Capacity was set at three at the beginning of the simulation, three optimized layup sequences are output at the end of the simulation. It was noticed that all three layup sequences led to the same end state as illustrated in Figure 7.20. The same five actions have been used to achieve the end state in all three cases. However, the sequence in which these actions have been used are different to each other. This is similar to the three solution paths explained for the preliminary case study in section 6.2. Numerical output statistics from the simulation on these solutions and action sequences are illustrated in Table 7.4. Refer to section 5.2.1 for details on action indexing. Essentially, all three solution paths involve one 'Down' action that brings in to contact 16 nodes of the net with the top flat surface of the mould and a combination of out of plane diagonal bending about the [-X, -Z] -> [X, Z] direction in upward (Action 16) and downward (Action 6) directions to bring in to contact 5 more nodes with the slanted faces and bottom flat surface of the mould.

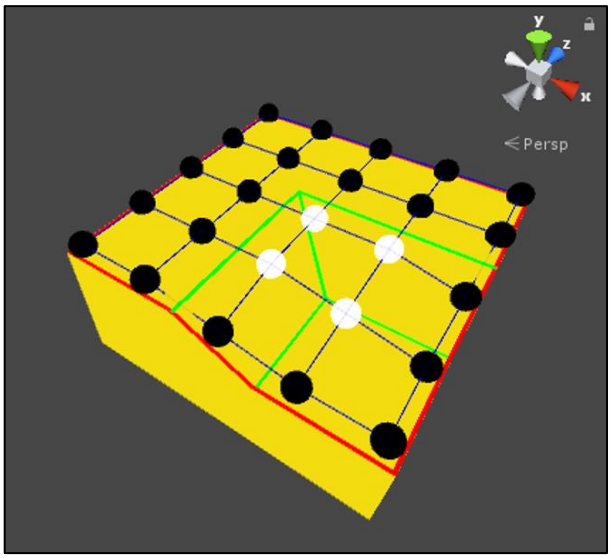


Figure 7.20 - End state (perspective view) obtained from the simulation

Table 7.4 - Numerical output statistics and action sequences from the simulation – case study 2

Solution No.	No. of Actions	Cumulative Reward	Action Sequence
1	5	0.84	Down, Action 16, Action 6, Action 16, Action 6
2	5	0.84	Down, Action 16, Action 16, Action 6, Action 6
3	5	0.84	Action 16, Down, Action 16, Action 6, Action 6

7.3.5 Understanding Drape Limitations

It is clear from Figure 7.20 that four nodes have failed to make contact with the mould. Recalling the layup sequence determined for this particular mould in section 4.2.2, the net will have to be pushed in to the groove to ensure that these four nodes gain contact with the mould. The simple model used for the deformation of the PJN in this simulation certainly does not allow the AI agent to take such an action. Therefore, the output from

this simulation cannot yet be used to produce layup instructions for complex mould shapes that require actions more complex than out of plane bending.

7.4 Summary

Working mechanics and limitations of the proposed artificially intelligent layup agent were explored in this chapter using two distinct case studies: a spherical mould and the 3D mould used for VR training in Chapter 4 of this thesis. Results from the case study involving the spherical mould suggested that the AI agent would fail to obtain the ideal action sequence required to complete the layup if relevant hyper-parameters were not fine-tuned to ideal values. It was also concluded that it is impossible to know what the ideal values for these hyper-parameters are at the beginning of the simulation. The straightforward solution to this issue is to set the relevant hyper-parameters to very small values. However, this could lead to unmanageable requirements in compute power. A possible solution to this issue was already suggested in section 6.5.

Results from the second case study suggested that the results from the proposed simulation, in its current state could not be used to produce layup instructions for complex mould shapes that require actions more complex than what the AI agent is currently capable of performing. This issue becomes very clear when comparing the set of actions the AI agent can take against the set of actions that were used in the VR training simulation in Chapter 4. In order to solve this issue, the set of actions available to the AI agent must be representative of actions taken by real world laminators. In [16], eight such distinct actions have been identified as illustrated in Figure 7.21. Simulating these eight actions would be a logical starting point. This is suggested as future work by the author.

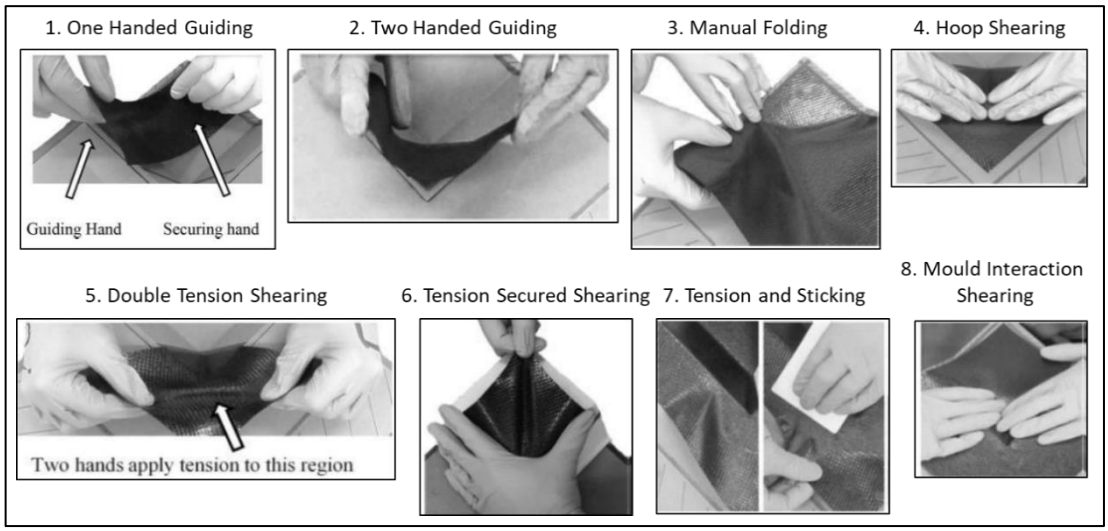


Figure 7.21 - Physical manipulations used by laminators as identified in [16]

In conclusion, the proposed artificially intelligent layup agent has provided a novel solution for predicting the ideal sequence of actions required to layup a composite mould. However, in its current state, it can only be used to create accurate layup instructions on very simple mould shapes, such as the one discussed in the preliminary case study. For more complex shapes, the simulator can be used as a layup guidance system, when used side-by-side with a composite laminator, especially with the use of action value predictions on distinct states, which could provide laminators with an idea of which actions are favourable and which are not, given a particular state.

CHAPTER 8: Conclusions on Current Research and Recommendations for Future Work

Although there exist many barriers in bridging the gap between technology and skills in the advanced composites industry, a potential workforce shortage, with the necessary skills to undertake the layup tasks, has been identified as one of the major concerns in the near term. Aligned to this is an existing and immediate need for re-skilling and/or up-skilling of the current workforce, as well as standardization of training delivery. Some countries, such as the UK have stated aims of doubling its composite workforce within the near future, although how such an increase in new workers (most likely from a variety of experiences and capabilities) can be quickly and efficiently trained remains unclear. A near chronic lack of access to resources and competent training (including evaluation processes) appears to have been a major issue to date, whether in the UK or in the off-shoring of parts overseas, leading to extended training schedules, excessive costs, owing to steep on-the-job learning curves and limited knowledge base capture/exploitation. These risks are further heightened by the possible escalation of composite use into other sectors, and increased product demand for those current sectors employing the material. Certainly, a mantra of 'Bigger, Faster, and Cheaper' has been adopted.

Hand layup has remained the dominant forming process in composite manufacturing due to the present inability of automated processes to form complex geometries, and mixed-material components requiring human interaction. Hand layup is a craft that is developed by skills and experience gained over a number of years and chosen process route and the final result can vary from laminator to laminator. Composite laminators have a choice and it is this choice and the lack of standardization in training delivery that leads to them developing their own set of techniques, skills, and methods of layup. Experience on layup techniques in general has been passed down directly from one laminator to the other, or gained through personal experience.

The delivery of laminator training is yet to be standardized, and mainly comes in the form of one-to-one tuition/coaching within the institution/company. Specialist training providers appear to be increasing in number but suffer the same standardization issues and limited flexibility to offer immediate on-site training. If this present situation is unsustainable then through the rising demands for composite products this problem will only become more acute. More adaptable and novel solutions must be found, and this PhD project has attempted to address these issues by exploiting Virtual Reality (VR) technology, the concept of Gamification and artificial intelligence (AI) techniques.

8.1 Conclusions and Future Work Suggestions on the use of VR Technology and Gamification for Skills Training within Hand Layup of Prepreg

A feasibility study on the use of VR technology and gamification on composites skills/knowledge transfer was carried out on the layup of a flat carbon fibre composite panel using unidirectional prepreg material in a typical composite clean room environment. The VR system used included a smartphone based head mounted display, hand held game pad as the input device and a headphone, which outputs audio instructions to the user. The VR simulation was designed to adopt the free-roam game genre with the user being placed in the virtual environment with a first person view. Steps to layup the carbon fibre flat panel were designed in to the VR simulation, and the user was forced to follow these steps in the order they were intended to be performed. Evaluation of the training aid was carried out in two stages. The first stage involved evaluating the training aid, using a pilot group in order to obtain feedback on technology acceptance and future improvements. Objective of the second evaluation stage was to test the effectiveness of the simulator as a knowledge/skills transfer tool on novice laminators (i.e. candidates who had no prior knowledge on composites or composite layup). The conclusion from the evaluation stages was that the designed VR training aid, in its current form is an effective knowledge capture/transfer tool. However, so far has limited use in physical skill transfer. As a knowledge capture/transfer tool, it has shown potential in the reduction of the learning curve of novice laminators as an increase in

task completion of greater than 20% was observed when test candidates used the designed VR training simulator as opposed to when candidates were just provided with video instructions. This specific method of delivery of training, in the form of head mounted VR gear was also well received by first time users, with only 6.7% of test candidates opting out during testing due to discomfort related issues, and the feedback received was mainly positive. It was also identified that this concept can easily be adapted to layup of more complex parts with the use of appropriate alternative hardware that would enable the transfer of physical skills via hand tracking.

With the understanding gained from the previous feasibility study, another VR training simulation was designed for a 3D mould shape. The VR system used for this simulation consisted of an Oculus Rift Development Kit 2 as the head mounted display and a Leap Motion Controller as a hand tracker, which allows hang gestures and movements to be programmed in to the VR simulation. Audio instructions were left out from this simulation, since layup instructions were provided in video form to the user through a layup instruction player embedded in the VR simulation. Nineteen distinct actions were identified as required to complete the layup of the chosen mould through video analysis of expert laminators laying up the chosen mould. Relevant experiments were carried out to obtain a preliminary understanding on the suitability of VR technology as a platform for hand lamination training. Overall feedback received by the test candidates suggest that the VR training aid in its current form may not be suitable as a standalone platform for hand lamination training, instead it is more suitable to be used in tandem with one-to-one training. However, experimental results did not fully agree with the opinions of the test candidates, since test candidates who received VR training only performed better (>20% increase in average accuracy score) than the test candidates who received one to one training during their layup trials. In addition to this, there was a >50% reduction in average time to completion of the VR training aid just after 3 trials at it. This means that even first time users of such a VR training aid can quickly grasp its working mechanics. It should also be noted that similar results would not have been obtained if a more complex mould shape that required complex manipulations to the prepreg (especially in plane shear) was used for the case study. This is because, in such

a case it would be critical for test candidates to get some form of haptic feedback during training that would give them an idea of how much force/pressure must be applied to the prepreg to obtain the required deformations. Since, the current VR training aid does not include any form of haptic feedback, it may not be suitable to train laminators on the layup of more complex shapes. Based on this understanding, the next logical step in this PhD project would have been to explore the use of haptic feedback in VR training simulations for hand lamination. However, the author decided not to go down this path for the rest of the duration of this PhD project, upon the realization of few major bottleneck in the concept of such VR training simulations as explained below. Instead, exploration of the use of haptic feedback is suggested as future work. VR force feedback gloves introduced by HaptX [148], VRgluv [149] and Plexus [150] could be potential exploration paths.

All deformations of the prepreg designed in to the VR simulation were pure animations that were predetermined to mimic as closely as possible the actual deformation of the prepreg. This means that the user does not have the opportunity to actively interact with the virtual PJN to try different layup sequences. The animation approach also means that each step of the VR simulation must be manually coded in to the VR simulation. This was found to be a very time consuming and an inefficient process, and there is very little room to automate the generation of VR training simulations for a given mould shape following this approach. The solution to these drawbacks would be the design of an interactive PJN that would deform actively based on the user input. In addition, currently there is no mechanism to decide which layup sequence is ideal for a given mould shape. Upon the realization of these limitations, the author decided to explore solutions to these limitations, instead of further exploring the use of VR technology for hand lamination training.

8.2 Conclusions and Future Work Suggestions on the Developed AI Layup Agent

Previous attempts made at using outputs from existing drape simulators to provide layup instructions were reviewed. The conclusion from this review was that such previous attempts have failed to provide sufficient information required for shop-floor operatives to carry out layup operations. In searching for a solution to this limitation of drape simulators, their fundamental approach was re-evaluated and it was concluded that perhaps the focus of drape simulators should shift more towards modelling the human learning process, coupled with a progressive deformation model for the reinforcement. The concept of reinforcement learning was introduced as a method to model the human learning process during hand layup. Reinforcement learning was defined as 'learning what to do – how to map states to actions – so as to maximize a numerical reward'. The concept of convolutional neural networks was then introduced and reviewed as a method of mapping states to actions in a reinforcement learning system. The theoretical foundation necessary to understand how the process of hand lamination of prepreg material will be modelled as a reinforcement learning system in order to address the issue of not being able to predict the ideal sequence of physical manipulations required to layup a composite part was provided.

With the necessary theoretical understanding in place, the modelling process for a preliminary case study on the use of computational reinforcement learning and convolutional neural networks to create a self-learning artificially intelligent composite layup agent was presented. The mould shape used for this preliminary case study was intentionally kept very simple in order to first understand the working mechanics of such a learning algorithm. The mould shape used in this case study contained a flat surface followed by a ramp at 45^{o}, which requires simple physical manipulation to a woven cloth to be laid up properly. The woven reinforcement was modelled as a rigid pin-jointed net. The layup process was modelled as a video game with a digital agent playing it. The agent is given the ability to perform a fixed number of actions in the form of manipulations to the net. A reward system has been put in place that awards the agent

with positive and negative rewards based on actions taken by the agent. The goal of the agent is to maximize the total cumulative reward it receives, while minimizing the number of total actions it performs.

Results from the preliminary case study suggested that the proposed artificially intelligent layup agent was successful in determining the ideal layup sequence for the mould shape it was presented. In addition, through a convolutional neural network, it was capable of building up an accurate representation of the reinforcement-learning environment, in the form of a mapping between states and action values. However, a multitude of hyper-parameters had to be fine-tuned to ensure the success of the AI agent. Furthermore, it was understood that many of the hyper-parameter values required to ensure the success of the AI agent are dependent on the mould shape and the resolution of the pin jointed net. This was identified as a major drawback in ensuring the success of the AI agent, when moulds with more complex geometries are in consideration.

The issue with determining ideal values for hyper-parameters, Rotational Constant and Translational Step may be solved by restricting node submersion in the mould. Achieving this would require a slightly different model for the interaction between the net and the mould as opposed to the model presented in this thesis. In such a model, the mould and the nodes making up the net would have solid surfaces and there would not exist any mechanism by which nodes could submerge in the mould. An example scenario of this concept applied to the 3D mould used in the VR training aid is illustrated in Figure 8.1. The exploration of such a model is suggested as future work.

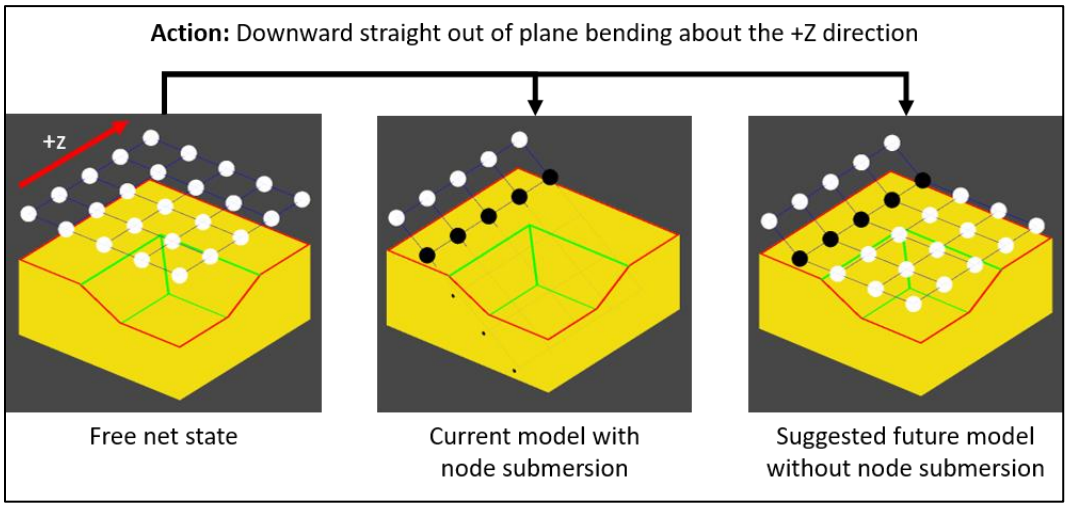


Figure 8.1 – Suggested interaction model to avoid node submersion

Results from a further case study carried out on the same 3D mould used for VR training suggested that the proposed AI layup agent, in its current state could not be used to produce layup instructions for complex mould shapes that require actions more complex than what the AI agent is currently capable of performing. This issue becomes very clear when comparing the set of actions, the AI agent can take against the set of actions that were used in the VR training simulation in Chapter 4. In order to solve this issue, the set of actions available to the AI agent must be representative of actions taken by real world laminators. Design of such a set of actions for the AI agent is also suggested as future work. In section 7.4, a starting point for such a set of actions was suggested in reference to [16]. This set of actions is illustrated once again in Figure 8.2 for the purpose of clarity of this section.

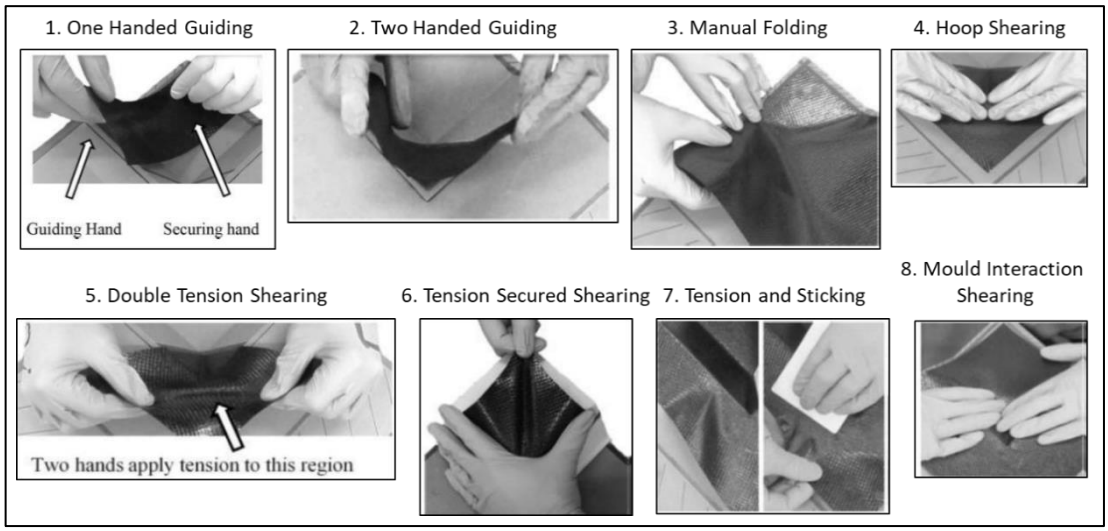


Figure 8.2 - Physical manipulations used by laminators as identified in [16]

8.3 Summary of Conclusions

- An existing and immediate need for re-skilling and/or up-skilling of the current workforce, as well as standardization of training delivery for composite laminators was identified, and Virtual Reality (VR) technology in tandem with the concept of Gamification was chosen as a potential solution to the identified issue.
- The smartphone-based VR training simulator designed for the layup of a flat prepreg panel is an effective knowledge capture/transfer tool. However, so far has limited use in physical skill transfer due to the lack of motion tracking.
- As a knowledge capture/transfer tool, it has shown potential in the reduction of the learning curve of novice laminators, as an increase in task completion of greater than 20% was observed when test candidates used the designed VR training simulator as opposed to when candidates were just provided with video instructions.
- The fact that only 6.7% of test candidates (2 out of 30), who were all first time VR users opted out during testing due to discomfort related issues illustrate the ease of adaptability and acceptance to this new technology for new users.
- The second VR training simulator, which was designed with integrated hand tracking allowed the simulation of layup of complex 3D mould shapes.
- Overall feedback received by the test candidates who were part of a preliminary evaluation study of this second simulator suggested that the VR training aid in its

current form may not be suitable as a standalone platform for hand lamination training, instead it is more suitable to be used in tandem with one-to-one training. This is due to the lack of haptic feedback in the current system.

- However, experimental results did not fully agree with the opinions of the test candidates, since test candidates who received VR training only performed better (>20% increase in average accuracy score) than the test candidates who received one to one training during their layup trials.
- In addition to this, there was a >50% reduction in average time to completion of the VR training aid just after 3 trials at it. This means that even first-time users of such a VR training aid can quickly grasp its working mechanics.
- In an attempt to automate the generation of VR simulations and determine ideal layup sequences, a reinforcement learning based artificially intelligent (AI) layup agent was introduced.
- Results from the preliminary case study of the AI agent suggested that the proposed AI layup agent was successful in determining the ideal layup sequence for the simple ramp mould shape it was presented with.
- In addition, through a convolutional neural network, it was capable of building up an accurate representation of the reinforcement-learning environment, in the form of a mapping between states and action values. However, a multitude of hyper-parameters had to be fine-tuned to ensure the success of the AI agent.
- Furthermore, it was understood that many of the hyper-parameter values required to ensure the success of the AI agent are dependent on the mould shape and the resolution of the pin jointed net. This was identified as a major drawback in ensuring the success of the AI agent, when moulds with more complex geometries are in consideration.
- Results from a further case study carried out on the same 3D mould used for VR training suggested that the proposed AI layup agent, in its current state could not be used to produce layup instructions for complex mould shapes that require actions more complex than what the AI agent is currently capable of performing.

8.4 Summary of Future Work

Future work suggested for VR training:

1. Adding haptic feedback: This is essential if VR training is to be used for more complex mould shapes where more complex manipulations such as in plane shear is required to form woven prepreg on to the mould without defects. This is because, in such a case, it would be critical for users to get some form of haptic feedback during training, that would give them an idea of how much force/pressure must be applied to the prepreg to obtain the required deformations.
2. Design of an interactive Pin Jointed Net (PJN): All deformations of the prepreg designed in to the final VR simulation were pure animations that were predetermined to mimic as closely as possible the actual deformation of the prepreg. This means that the user does not have the opportunity to actively interact with the virtual PJN to try different layup sequences. The animation approach also means that each step of the VR simulation must be manually coded in to the VR simulation. This was found to be a very time consuming and an inefficient process and there is very little room to automate the generation of VR training simulations for a given mould shape following this approach. The solution to these drawbacks would be the design of an interactive PJN that would deform actively based on the user input.

Future work suggested for the AI layup agent:

1. Improvements to the interaction model between the PJN and the mould: It was identified that determining ideal hyper parameters was essential for the success of the AI layup agent. This issue may be solved by restricting node submersion in the mould. This could be achieved with a model where the mould and the nodes making up the net would have solid surfaces and there would not exist any mechanism by which nodes could submerge in the mould.
2. Increasing the depth of actions available to the AI layup agent: The inability of the AI agent to layup the 3D mould used in the VR training simulator suggested that the set

of actions available to the AI agent must be more representative of actions taken by real world laminators.

8.5 Revisiting the Research Question and Understanding the Impact of the Research Presented

In the 1st chapter of this thesis, the research question was presented as:

"Given a completely new mould geometry, how can we provide standardized training to multiple laminators around the world in a fast and economical manner, so that the final layups are identical to each other?"

Prior to the research presented in this thesis, the only available methods for delivery of training were:

1. One-to-one training
2. Self-learning

Both the above methods were not going to lead to worldwide standardization of training delivery, due to the significant human factor involved in those methods. Research presented in this thesis has introduced a novel platform in the form of VR and motion tracking technologies to perform standardized training for laminators. Compared to currently available methods, VR training could lead to worldwide standardization of training delivery. Through preliminary case studies that yielded a >20% increase in accuracy of layup, the effectiveness of such a platform has been proven.

In addition to this, currently available training methods are also extremely time consuming and costly due to their large dependency on manual intervention. The proposed VR training platform is expected to significantly reduce training costs and speed up the training process due to the following reasons:

1. VR training is completely independent. This means that an expert laminator is not required to train the novice.

2. Material wastage during training is minimized due to the training process not requiring any prepreg material.
3. Elimination of the requirement for clean room conditions to carry out training means that training can be carried out anywhere and anytime with minimal pre preparation.
4. Large groups of novice laminators can be trained simultaneously due to the ease of reproducing training material, due to its digital nature.

Moreover, prior to the research presented in this thesis, there was no available method to produce standardized and numerically verified manufacturing instruction sheets for hand laminators. The AI layup agent introduced in this thesis has opened up a completely new research avenue for the development of future drape simulators that will be capable of providing optimized and numerically validated instructions for hand laminators and the design of VR training simulations.

References

[1] D. Hull and T. Clyne, An Introduction to Composite Materials, Cambridge University Press, 1996.

[2] “Composites: Airbus continues to shape the future,” Airbus, [Online]. Available: http://www.airbus.com/newsroom/news/en /2017/08/composites--airbus-continues-to-shape-the-future.html. [Accessed 12 10 2017].

[3] “AERO - Boeing 787 from the Ground Up,” Boeing, [Online]. Available: http://www.boeing.com/commercial/aeromagazine/articles/qtr_4_06/ article_04_2.html. [Accessed 12 10 2017].

[4] “How Is Carbon Fiber Used to Make Formula 1 Cars?,” ThoughtCo, [Online]. Available: https://www.thoughtco.com/ formula1-car-as-carbon-fiber-cake-1347093. [Accessed 15 08 2017].

[5] “Reducing CO2 emissions from passenger cars - Climate Action - European Commission,” Climate Action - European Commission, [Online]. Available: https://ec.europa.eu/clima/policies/transport/vehicles/cars_en. [Accessed 12 01 2018].

[6] “Kyoto Protocol to the united nations framework convention on climate change,” United Nations, [Online]. Available: http://unfccc.int/resource/docs/convkp/kpeng.pdf. [Accessed 12 01 2018].

[7] “The City Mobility Concept,” BMW, [Online]. Available: http://www.bmw.co.uk/en_GB/new-vehicles/bmw-i/bmw-i/philosophy.html. [Accessed 13 01 2018].

[8] M. Freitag, “Carbon - BMW hat Nachschubprobleme beim Bau des i3,” manager magazin, [Online]. Available: http://www.manager-magazin.de/unternehmen/autoindustrie/ carbon-bmw-hat-nachschubprobleme-beim-bau-des-i3-a-954274.html. [Accessed 12 01 2018].

[9] “Production forecast for the Ford Focus 2017 | Statistic,” Statista, [Online]. Available: https://www.statista.com/statistics/202769/worldwide-production-forecast-of-the-ford-focus/. [Accessed 12 01 2018].

[10] D. Lukaszewicz, C. Ward and K. Potter, “The engineering aspects of automated prepreg layup: History, present and future,” *Composites Part B: Engineering,* vol. 43, pp. 997-1009, 2012.

[11] B. T. Astrom, Manufacturing of Polymer Composites, Cheltenham: Nelson Thornes, 2002.

[12] S. Advani and K. Hsiao, Manufacturing techniques for polymer matrix composites, Cambridge: Woodhead Publishing Limited, 2012.

[13] M. Bader, W. Smith, A. Isham and J. Rolston, Delaware Composites Design Encyclopaedia. Vol. 3: Processing and Fabrication Technology, Pennsylvania USA: Technomic Publishing Company Inc, 1990.

[14] F. Campbell, Unique materials that require unique processes, Elsevier Science, 2003.

[15] S. V. Hoa, Principles of the manufacturing of composite materials, Pennsylvania, USA: DEStech Publications, Inc, 2009.

[16] M. Elkington, D. Bloom, C. Ward, A. Chatzimichali and K. Potter, "Hand layup: understanding the manual process," *Advanced Manufacturing: Polymer & Composites Science,* pp. 138-151, 2015.

[17] P. J. Schubel, "Cost modelling in polymer composite applications: Case study–Analysis of existing and automated manufacturing processes for a large wind turbine blade," *Composites Part B: Engineering,* vol. 43(3), pp. 953-960, 2012.

[18] R. Mezzacassa, V. Callado and F. J. estensoro, "Fully Automated energy efficient 3D Preforming. Tecnalia Research and Inovation," *JEC Magazine 77,* pp. 48-50, 2012.

[19] K. Kendall, C. Rudd, M. Owen and V. Middleton, "Characterization of the resin transfer moulding process," *Composites Manufacturing,* vol. 3(4), pp. 235-241, 1992.

[20] J. Lightfoot, M. Wisnom and K. Potter, "Defects in woven preforms: Formation mechanisms and the effects of laminate design and layup protocol," *Composites Part A: Applied Science and Manufacturing,* vol. 51, pp. 99-107, 2013.

[21] K. Potter, B. Khan, M. Wisnom, T. Bell and J. Stevens, "Variability, fibre waviness and misalignment in the determination of the properties of composite materials and structures," *Composites Part A: Applied Science and Manufacturing,* vol. 39(9), pp. 1343-1354, 2008.

[22] M. Elkington, The evolution and automation of sheet prepreg layup, PhD Thesis: University of Bristol, 2015.

[23] F. Nezami, T. Gereke, M. Hubner, O. Dobrich and C. Cherif, "Factors of process robustness in multilayer preforming of carbon fibre reinforcements," in *16th European Conference on Composite Materials, University of Seville*, Seville, 2014.

[24] D. Bender, J. Schuster and D. Heider, "Flow rate control during vacuum-assisted resin transfer molding (VARTM) processing," *Composites Science and Technology,* vol. 66(13), pp. 2265-2271, 2006.

[25] N. Correia, F. Robitaille, A. Long, C. Rudd, P. Simacek and S. Advani, "Use of resin transfer molding simulation to predict flow, saturation, and compaction in the VARTM process," *Journal of fluids engineering,* vol. 126(2), pp. 210-215, 2004.

[26] N. Naik, M. Sirisha and A. Inani, "Permeability characterization of polymer matrix composites by RTM/VARTM," *Progress in Aerospace Sciences,* vol. 65, pp. 22-40, 2014.

[27] O. Faruk and M. S. Ain, "Biofiber reinforced polymer composites for structural applications," in *Developments in Fiber-Reinforced Polymer (FRP) Composites for Civil Engineering*, Cambridge, UK, Woodhead Publishing, 2013, pp. 18-53.

[28] C. Acquash, “Optimization under uncertainty of a composite fabrication process using a deterministic one-stage approach,” *Computers & Chemical Engineering,* vol. 30, no. 6-7, pp. 947-960, 2006.

[29] “Composites Manufacturing – Aerospace Engineering Blog,” Aerospace Engineering Blog, 12 07 2012. [Online]. Available: https://aerospaceengineeringblog.com/composite-manufacturing/. [Accessed 26 03 2019].

[30] M. Y. Matvee, P. J. Schubel, A. C. Long and I. A. Jones, “Understanding the buckling behaviour of steered tows in Automated Dry Fibre Placement (ADFP),” *Composites Part A: Applied Science and Manufacturing,* vol. 90, pp. 451-456, 2016.

[31] Z. Gurdal and R. Olmedo, “Inplane response of laminates with spatially varying fiber orientations – variable stiffness concept,” *AIAA J,* vol. 31, no. 4, pp. 751-758, 1993.

[32] Z. Gurdal, B. F. Tatting and C. K. Wu, “Variable stiffness composite panels: effects of stiffness variation on the in-plane and buckling response,” *Composites Part A: Applied Science and Manufacturing,* vol. 39, no. 5, pp. 911-922, 2008.

[33] B. C. Kim, K. Hazra, P. Weaver and K. Potter, “Limitations of fibre placement techniques forvariab;e ang;e tow composites and theor process induced defects,” in *18th international conference on composites materials*, Korea, 2011.

[34] G. Marsh, “Automating aerospace composites production with fibre placement,” *Reinforced Plastics,* vol. 55, no. 3, pp. 32-37, 2011.

[35] S. Edwards and D. Crowley, “Augmented learning for high dexterity manufacturing,” Ufi Charitable Trust, 2018.

[36] D. Crowley, C. Ward and K. Potter, “A status of acceptance criteria and process requirements in advanced composites manufacturing, and whether they are fit for purpose,” SAE Technical Paper, 2013.

[37] D. Gay and S. V. Hoa, Composite Materials: Design and Applications, CRC Press, 2006.

[38] P. Teixeira-Lage, “Composite design challenges: Airbus a350xwb wing covers,” *JEC composites no 89,* pp. 37-39, 2014.

[39] M. Morell, “Composites in the Aerospace Sector: Technological Excellence through Innovation as key factor for Competitiveness — the A400M,” in *Plenary Lecture III: 16th European Conference on Composite Materials, University of Seville*, Seville, 2014.

[40] Aviation Week, “Technology to Raise Build Rates, Lower Costs,” [Online]. Available: http://aviationweek.com/awin/technology-raise-build-rates-lower-costs. [Accessed 16 01 2018].

[41] C. Calladine, Theory of shell structures, Cambridge: Cambridge University Press, 1983.

[42] M. Elkington, D. Bloom, C. Ward, A. Chatzimichali, and K. Potter, "Understanding the Lamination Process," in *Proceedings of the 19th International conference on composite materials*, 2013.

[43] K. Potter, "The influence of accurate stretch data for reinforcements on the production of complex structural mouldings: Part 1. Deformation of aligned sheets and fabrics," *Composites,* vol. 10, no. 3, pp. 161-167, 1979.

[44] M. Kaufmann, D. Zenkert and M. Akermo, "Cost/weight optimization of composite prepreg structures for best draping strategy," *Composites Part A: Applied Science and Manufacturing,* vol. 41, pp. 464-472, 2010.

[45] R. Tatlock, "Manufacturing Process Automation," BioProcess International, [Online]. Available: http://www.bioprocessintl.com/manufacturing/information-technology/manufacturing-process-automation-325684/. [Accessed 25 10 2017].

[46] M. Elkington, C. Ward and A. Sarkytbayev, "Automated composites draping: A review," in *SAMPE North America*, Seattle, 2017.

[47] H. V. Jones, A. P. Chatzimichali, R. Middleton, K. D. Potter and C. Ward, "Exploring the discrete tools used by laminators in composites manufacturing: application of novel concept," *Advanced Manufacturing: Polymer & Composites Science,* vol. 1:4, pp. 185-198, 2015.

[48] D. Bloom, M. Elkington, C. Ward, A. Chatzimichali and K. Potter, "On prepreg properties and manufacturability," in *The 19th International Conference on Composite Materials*, Montreal, 2013.

[49] M. Such, C. Ward, W. Hutabarat and A. Tiwari, "Intelligent Composite Layup by the Application of Low Cost Tracking and Projection Technologies," *Procedia CIRP 25,* p. 122–131, 2014.

[50] D. Crowley and C. Ward, "Augmenting Composites Layup Training," LayupRITE , [Online]. Available: https://layuprite.blogs.bristol.ac.uk/. [Accessed 26 03 2019].

[51] S. G. Hancock, Forming woven fabric reinforced composite materials for complex shaped components: informing manufacture with virtual prototyping, PhD Thesis: University of Bristol, 2006.

[52] S. Kularatna, "Unity woven prepreg drape simulator," 2019. [Online]. Available: https://shashitha-kularatna.itch.io/unity-drape-simulator. [Accessed 09 04 2019].

[53] S. G. Hancock and K. D. Potter, "The use of kinematic drape modelling to inform the hand lay-up of complex composite components using woven reinforcements," *Composites Part A,* vol. 37 , no. 3, pp. 413-422, 2006.

[54] S. G. Hancock and K. D. Potter, "Inverse drape modelling - an investigation of the set of shapes that can be formed from continuous aligned woven fibre reinforcements," *Composites Part A,* vol. 36 , no. 7, pp. 947-953, 2005.

[55] C. Ward, S. Hancock and K. Potter, “Forming complex shaped components using drape simulation software: informing manual and automated production needs,” *SAE International Journal of Aerospace,* vol. 1, no. 1, pp. 798-810, 2008.

[56] S. Kularatna, “Gamified Training Aid for Composite Layup: Using Low Cost Smartphone Based Virtual Reality,” *ACCIS (Internal unpublished document),* 2014.

[57] “The UK Composites Strategy. (2009). 1st ed. [eBook],” The Department for Business Innovation and Skills (UK), [Online]. Available: http://www.ncn-uk.co.uk/uploads/UKCompositesStrategy.pdf. [Accessed 19 03 2015].

[58] “The UK composites industry: bridging the gap between technology and skills - Reinforced Plastics,” Elsevier Ltd, U, 2015. [Online]. Available: http://www.reinforcedplastics.com/view/31758/the-uk-composites-industry-bridging-the-gap-between-technology-and-skills. [Accessed 19 03 2015].

[59] P. Lewis, “Skills and training for composites manufacturing and the use in the UK: An analysis,” [Online]. Available: http://www.gatsby.org.uk/uploads/education/reports/pdf/composites-final.pdf. [Accessed 17 01 2018].

[60] P. Lewis, “Flying high? A study of technician duties, skills, and training in the UK aerospace industry,” Gatsby Charitable Foundation, 2012.

[61] D. Crowley, “A study into composite laminators’ motivation,” in *FAIM 2013*, Porto, PT. Universidad do Porto, 2013.

[62] “CompositesUK, Training Courses in Composite Materials,” 2018. [Online]. Available: https://compositesuk.co.uk/composite-materials/education/training-courses. [Accessed 01 02 2019].

[63] “EAL, Composite Engineering Level 2,” 2018. [Online]. Available: http://eal.org.uk/record/2342-composite-engineering-qcf. [Accessed 01 04 2019].

[64] “EAL, Composite Engineering Level 3,” 2018. [Online]. Available: http://eal.org.uk/record/2335-composite-engineering-qcf. [Accessed 01 04 2019].

[65] B. Thornton and D. Jones, “Composites technician trailblazer apprenticeship,” 2015. [Online]. Available: http://www.nfec.org.uk/regions/north-east/Composite%20trailblazer%20presentation.pdf. [Accessed 09 04 2019].

[66] “Composites Assured Practitioner (CAP) scheme,” Hemel Hempstead, UK:, [Online]. Available: https://compositesuk.co.uk/industry-support/composites-assuredpractitioner-cap-scheme. [Accessed 2018 01 17].

[67] “A Study into the Status, Opportunities and Direction for the UK Composites,” Composites UK, 2013. [Online]. Available: https://compositesuk.co.uk/system/files/documents/CLF%20UK%20Composites%20Study%202013%20Summary.pdf. [Accessed 18 01 2018].

[68] C. Ward, K. Hazra and K. Potter, “Development of the manufacture of complex composite panels,” *International Journal of Materials and Product Technology,* vol. 42, p. 131, 2013.

[69] G. Campos, “Lifelike VR surgery training to cost ‘less than a dead body’,” 2018. [Online]. Available: https://www.avinteractive.com/news/virtual-augmented-mixed/simulation-service-launch-aimed-surgeons-21-08-2018/. [Accessed 09 04 2019].

[70] N. Zhou and Y. Deng, “Virtual reality: A state-of-the-art survey,” *International Journal of Automation and Computing,* vol. 6, no. 4, pp. 319-325, 2009.

[71] G. Keenaghan, “State of the Art of Using Virtual Reality Technologies in Built Environment Education,” in *TMCE*, Budapest, 2014.

[72] “Technology | Jaguar Land Rover Corporate Website,” Jaguarlandrover.com, 2015. [Online]. Available: http://www.jaguarlandrover.com/gl/en/innovation/technology/. [Accessed 19 03 2015].

[73] D. Szondy, “Factory of the future,” Airbus Group, [Online]. Available: http://www.airbusgroup.com/int/en/story-overview/factory-of-the-future.html. [Accessed 19 03 2015].

[74] “Applications of Virtual Reality,” Virtual Reality Society, [Online]. Available: http://www.vrs.org.uk/virtual-reality-applications/index.html. [Accessed 19 03 2015].

[75] P. Milgram, H. Takemura, A. Utsumi and F. Kishino, “Augmented Reality: A class of displays on the reality-virtuality continuum,” *International society for optics and photonics,* vol. 2351, pp. 282-292, 1995.

[76] T. Mikropoulos and A. Chalkidis, “Students' Attitudes Towards Educational Virtual Environments,” *Education and Information Technologies,* vol. 3, no. 2, pp. 137-148, 1998.

[77] V. Pantelidis, “Reasons to Use Virtual Reality in Education and Training Courses and a Model to Determine When to Use Virtual Reality,” *Themes in Science and Technology Education,* vol. 2, pp. 59-70, 2009.

[78] R. Aked, T. Fernando, N. Murray and G. Gautier, “State of the art in VR,” 2007. [Online]. Available: http://www.cs.upc.edu/~virtual/SGI/docs/3.%20Further%20Reading/State-of-the-art%20in%20VR%20-%20Intuition.pdf. [Accessed 16 02 2016].

[79] D. Bowman and R. McMahan, “Virtual Reality: How Much Immersion Is Enough?,” *Computer,* vol. 40, no. 7, pp. 36-43, 2007.

[80] J. Ripton and L. Prasuethsut, “The VR race: What you need to know about Oculus Rift, HTC Vive and more,” Techradar, [Online]. Available: http://www.techradar.com/news/world-of-tech/future-tech/the-vr-race-who-s-closest-to-making-vr-a-reality--1266538/2. [Accessed 19 03 2015].

[81] “Oculus,” Oculus.com, 2018. [Online]. Available: https://www.oculus.com/. [Accessed 2018 01 18].

[82] “Oculus Go,” Oculus.com, 2018. [Online]. Available: https://www.oculus.com/go/. [Accessed 18 01 2018].

[83] “VIVE™ | Discover Virtual Reality Beyond Imagination,” Vive.com, 2018. [Online]. Available: https://www.vive.com/uk/. [Accessed 18 01 2018].

[84] “HMD_PC Computing Accessories - XE800ZAA-HC1US | Samsung US,” Samsung Electronics America, 2017. [Online]. Available: https://www.samsung.com/us/computing/hmd/windows-mixed-reality/xe800zaa-hc1us-xe800zaa-hc1us/. [Accessed 18 01 2018].

[85] A. Steed and S. Julier, “Design and implementation of an immersive virtual reality system based on a smartphone platform,” in *IEEE Symposium on 3D User Interfaces 2013*, Orlando, 2013.

[86] M. Aono, D. Breen and M. Wozny, “Modeling methods for the design of 3D broadcloth,” *Composite Parts, Computer Aided Design,* vol. 33(13), pp. 989-1007, 2001.

[87] S. Sharma, P. Potluri and I. Porat, “Moulding analysis of 3D woven composite preforms, mapping algorithms,” in *12th international conference on composite materials*, Paris, 1999.

[88] C. Rudd, A. Long, P. McGeehin, F. Cucinella and L. Bulmer, “Processing and mechanical properties of bi-directional preforms for liquid composite moulding,” *Composite Manfacture,* vol. 6, p. 211, 1995.

[89] X. Yu, L. Zhang and Y. Mai, “Modelling and finite element treatment of intra-ply shearing of woven fabric,” *Journal of Materials Processing Technology,* vol. 138, pp. 47-52, 2003.

[90] B. Boubaker, B. Haussy and J. Ganghoffer, “Discrete models of fabrics accounting for yarn interactions,” *Europian Journal of Computational Mechanics,* vol. 14, pp. 653-676, 2005.

[91] D. Orgeas, S. Corre and D. Favier, “Anisotropic viscous behaviour of sheet moulding compounds (SMC) during compression moulding,” *International Journal of Plasticity,* vol. 19, pp. 625-646, 2003.

[92] P. Boisse, B. Zouari and A. Gasser, “A mesoscopi approach for the simulation of woven fibre composite forming,” *Composites Science and Technology ,* vol. 65, pp. 429-436, 2005.

[93] V. Maymard, “Master of Science Thesis: A new approach to simulating the draping of prepreg composites manufactured by hand,” Stockholm, Sweden, 2017.

[94] R. Sutton and A. Barto, Reinforcement Learning: An Introduction, CAmbridge, Massachusetts, London, England: The MIT Press, 2012.

[95] C. Diuk, A. Cohen and M. Littman , “An object-oriented representation for efficient reinforcement learning,” in *International Conference on Machine Learning*, 2008.

[96] “Overview and Applications of Artificial Neural Networks,” Medium, [Online]. Available: https://medium.com/@xenonstack/overview-of-artificial-neural-networks-and-its-applications-2525c1addff7. [Accessed 13 02 2018].

[97] S. Haykin, Neural Networks and Learning Machines, New Jersey: Pearson Prentice Hall, 1999.

[98] “What is the vanishing gradient problem?,” Quora, [Online]. Available: https://www.quora.com/What-is-the-vanishing-gradient-problem. [Accessed 14 02 2018].

[99] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[100] A. Karpathy, “CS231n Convolutional Neural Networks for Visual Recognition,” Cs231n.github.io, [Online]. Available: http://cs231n.github.io/neural-networks-1/. [Accessed 23 02 2018].

[101] A. Sharma, “Understanding Activation Functions in Deep Learning | Learn OpenCV,” Learnopencv.com, [Online]. Available: https://www.learnopencv.com/ understanding-activation-functions-in-deep-learning/. [Accessed 23 02 2018].

[102] S. Becker, “Unsupervised learning procedures for neural networks,” *International Journal,* vol. 2, pp. 17-33, 1991.

[103] L. Yann, “RI Seminar: Yann LeCun : The Next Frontier in AI: Unsupervised Learning,” YouTube, [Online]. Available: https://www.youtube.com/watch?v=IbjF5VjniVE. [Accessed 15 02 2018].

[104] G. Marcus, “In defense of skepticism about deep learning – Gary Marcus – Medium,” Medium, 14 01 2018. [Online]. Available: https://medium.com/@GaryMarcus/in-defense-of-skepticism-about-deep-learning-6e8bfd5ae0f1. [Accessed 2018 02 15].

[105] Z. Ghahramani, “Unsupervised Learning,” Gatsby Computational Neuroscience Unit, University College London, UK, [Online]. Available: http://www.inf.ed.ac.uk/teaching/courses/pmr/docs/ul.pdf. [Accessed 15 02 2018].

[106] J. Karhunen and T. Raiko, “Unsupervised Deep Learning: A Short Review,” Dept. of Information and Computer Science, Aalto University, Espoo, Finland, [Online]. Available: https://users.ics.aalto.fi/praiko/papers/karhunen2015.pdf. [Accessed 15 02 2018].

[107] “CS231n Convolutional Neural Networks for Visual Recognition,” Cs231n.github.io, [Online]. Available: http://cs231n.github.io/convolutional-networks/. [Accessed 15 02 2018].

[108] J. Springenberg , “Striving for Simplicity: The All Convolutional Net,” in *ICLR*, San Diego, 2015.

[109] S. Legg and M. Hutter, “Universal intelligence: a definition of machine intelligence,” *Minds Mach,* vol. 17, pp. 391-444, 2007.

[110] M. Genesereth, N. Love and B. Pell, “General game playing: overview of the AAAI,” *AI Magazine,* vol. 26, pp. 62-77, 2005.

[111] M. Bellemare, J. Veness and M. Bowling, “Investigating contingency awareness using Atari 2600 games,” in *Conference AAAI. Artificial Intelligence*, 2012.

[112] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning,* vol. 2, pp. 1-127, 2009.

[113] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems,* vol. 25, pp. 1106-1114, 2012.

[114] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science,* vol. 313, pp. 504-507, 2006.

[115] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” in *IEEE*, 1998.

[116] V. Mnih and a. et, “Human-level control through deep reinforcement,” *Nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[117] D. Hubel and T. Wiesel, “Shape and arrangement of columns in cat's striate cortex,” *journal of Physiology ,* vol. 165, pp. 559-568, 1963.

[118] I. Chorley, Interviewee, *The layup of a prepreg flat panel.* [Interview]. 05 02 2014.

[119] S. Kularatna, “VR Simulation - Layup of a flat prepreg panel,” [Online]. Available: https://www.youtube.com/watch?v=ZC6RsEJi0fM&t=5s. [Accessed 19 01 2018].

[120] S. Kularatna, “Action sequence for the layup of a uni directional prepreg flat panel,” [Online]. Available: https://www.youtube.com/watch?v=Eb2Qtiuj3zM. [Accessed 2019 04 08].

[121] C. Nutt, “How John Carmack is bending Samsung's VR strategy,” Gamasutra.com, [Online]. Available: https://www.gamasutra.com/view/news/226112/How_John_Carmack_is_bending_Samsungs_VR_strategy.php. [Accessed 19 03 2015].

[122] Unity Technologies, “Unity - Manual: Optimizing graphics performance,” [Online]. Available: https://docs.unity3d.com/Manual/OptimizingGraphicsPerformance.html. [Accessed 28 11 2014].

[123] S. Balk, M. Bertola and V. Inman, “Simulator Sickness Questionnaire: Twenty Years,” in *Seventh International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, New York, 2013.

[124] Student, “The probable error of a mean,” *Biometrika,* vol. 6, no. 1, p. 1, 1908.

[125] “Oculus Rift: A Look at Oculus VR's Flagship Virtual Reality Headset,” Lifewire, [Online]. Available: https://www.lifewire.com/oculus-rift-4157688. [Accessed 02 04 2018].

[126] A. Dobra, “New and Improved Oculus Rift Dev Kit 2 Revealed, Ships in July,” softpedia, [Online]. Available: http://news.softpedia.com/news/ New-and-Improved-Oculus-Rift-Dev-Kit-2-Revealed-Ships-in-July-433129.shtml. [Accessed 02 04 2018].

[127] A. Colgan, “How Does the Leap Motion Controller Work?,” Leap Motion Blog, [Online]. Available: http://blog.leapmotion.com/ hardware-to-software-how-does-the-leap-motion-controller-work/. [Accessed 02 04 2018].

[128] S. Kularatna, “VR Simulation - 3D Mould,” [Online]. Available: https://www.youtube.com/watch?v=rDhUAkzPN0E. [Accessed 2018 04 17].

[129] R. Likert, “A Technique for the Measurement of Attitudes,” *Archives of Psychology,* vol. 140, pp. 1-55, 1932.

[130] “Wireless VR | Unleash the VR World – TPCast,” [Online]. Available: https://www.tpcastvr.com/product. [Accessed 2018 04 11].

[131] “Unity3D,” [Online]. Available: https://unity3d.com/. [Accessed 16 02 2018].

[132] C. Mack and M. Taylor, “The Fitting of Woven Cloth to Surfaces,” *Journal of the Textile Institute Transactions,* pp. 477-488, 1956.

[133] K. Potter, “Beyond the pin-jointed net: maximising the deformability of aligned continuous fibre reinforcements,” *Composites Part A: Applied Science and Manufacturing,* vol. 33, no. 5, pp. 677-686, 2002.

[134] A. Sharma, M. Sutcliffe and S. Chang, “Characterisation of material properties for draping of dry woven composite material,” *Composites: Part A,* vol. 34, pp. 1167-1175, 2003.

[135] C. Lai and W. Young, “Modelling the fibre slippage during preforming of woven fabrics,” Department of Aeronautics and Astronautics National Cheng-Kung University , Tainan, Taiwan, 2018.

[136] A. C. Long, Design and Manufacture of Textile Composites, Cambridge: Woodhead, 2005.

[137] D. Mishkin and J. Matas, “All you need is a good init,” *arXiv preprint arXiv:1511.06422,* 2015.

[138] A. Karpathy, “CS231n Convolutional Neural Networks for Visual Recognition,” Cs231n.github.io, [Online]. Available: http://cs231n.github.io/neural-networks-2/#init. [Accessed 23 02 2018].

[139] X. Glorot and Y. Bengio , “Understanding the difficulty of training deep feedforward neural networks,” in *Thirteenth International Conference on Artificial Intelligence and Statistics*, Sardinia, Italy, 2010.

[140] K. He, X. Zhang, S. Ren and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *Microsoft Research,* 2015.

[141] M. Mazur, “A Step by Step Backpropagation Example,” [Online]. Available: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/. [Accessed 24 02 2018].

[142] M. Moreira and E. Fiesler, “Neural Networks with Adaptive Learning Rate and Momentum Terms,” IDIAP Research Institute, 1995.

[143] J. Duchi, E. Hazan and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” vol. 12, pp. 2121-2159, 2011.

[144] D. Kingma and J. Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” in *ICLR*, San Diego, 2015.

[145] G. Hinton, N. Srivastava and K. Swersky, “Neural Networks for Machine Learning: Lecture 6a - Overview of mini-batch gradient descent,” Cs.toronto.edu, [Online]. Available: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf. [Accessed 24 02 2018].

[146] N. Qian, “On the momentum term in gradient descent learning algorithms. Neural Networks,” *The Official Journal of the International Neural Network Society,* vol. 12, no. 1, pp. 145-151, 1999.

[147] S. Allaoui, “Experimental preforming of highly double curved shapes with a case corner using an interlock reinforcement,” *International Journal of Material Forming,* vol. 7, no. 2, pp. 155-165, 2012.

[148] S. Varga, “Haptic gloves for VR training, simulation, and design,” HaptX, [Online]. Available: https://haptx.com/. [Accessed 12 04 2019].

[149] “Force Feedback Gloves for VR Training,” VRgluv , [Online]. Available: https://www.vrgluv.com/. [Accessed 12 04 2019].

[150] P. Corp, “High-performance VR/AR Gloves,” Plexus, [Online]. Available: http://plexus.im/. [Accessed 12 04 2019].

[151] M. Meder and B. Jain, “The Gamification Design Problem,” in *Human-Computer Interaction*, Cornell University Library, 2014.

[152] S. Deterding, D. Dixon, R. Khaled and L. Nacke, “From game design elements to gamefulness: defining "gamification",” in *15th International Academic MindTrek Conference: Envisioning Future Media Environments*, New York, 2011.

[153] K. Wilson, W. Bedwell, E. Lazzara, E. Salsa, C. Burke, J. Estok, K. Orvis and C. Conkey, “Relationships Between Game Attributes and Learning Outcomes: Review and Research Proposals,” *Simulation & Gaming,* vol. 40, no. 2, pp. 217-266, 2008.

[154] M. Arbib, Brains, Machines, and Mathematics, 2nd edition, New York: Springer-Verlag, 1987.

[155] C. Ramón, Histologie du Systéms Nerveux de l'homme et des vertébrés, Paris: Maloine, 1911.

[156] G. Shepherd, The Synoptic Organization of the Brain, 3rd ed, New York: Oxford University Press, 1990.

[157] W. Freeman, Mass Action in the Nervous System, New York: Academic Press, 1975.

[158] "File:Neuron.svg - Wikimedia Commons," Commons.wikimedia.org, [Online]. Available: https://commons.wikimedia.org/wiki/File:Neuron.svg. [Accessed 13 02 2018].

[159] P. Churchland and T. Sejnowski, The Computational Brain, Cambridge: MA:MIT Press, 1992.

[160] A. Brodal, Neurological Anatomy in Relation to Clinical Medicine, 3rd ed, New York: Oxford University Press, 1981.

[161] D. Silver and et al, "Mastering the game of Go without human knowledge," *Nature,* vol. 19, no. 24270, pp. 354-359, 2017.

[162] J. Tsitsiklis and B. Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control,* vol. 165, pp. 674-690, 1997.

Appendix

Appendix: Table 1 - Understanding research motivations of each chapter of this thesis

Chapter Number	Research Motivations
2	This chapter is the result of a technology survey, which was carried out at the beginning of this PhD, with the objective of identifying new technologies that could serve as media to provide low cost, fast and standardized training to hand laminators. Virtual Reality (VR) technology was identified as the most suitable medium to meet this objective. Therefore, in this chapter, through a literature review, a case is made on why VR technology might be suitable to meet this objective.
3	Upon identifying VR technology as a suitable medium to meet research objectives of this PhD, a feasibility study was carried out to verify this claim. The process chosen for the feasibility study is the layup of a flat carbon fibre composite panel using unidirectional prepreg material in a typical composite clean room environment.
4	The preliminary case study carried out in Chapter 3 suggested that if VR technology is to be used as a platform for training hand laminators, especially when it comes to 3D complex mould shapes, hand tracking is a necessity. Therefore, a case study was carried out with the use of VR and hand tracking technologies for a 3D mould shape.
5	Upon completing the case study in chapter 4, it was identified that one of the major bottlenecks in designing Virtual Reality training aids or any other form of manufacturing instruction set for hand laminators in laying up complex composite parts is the inability to predict the best order of

Chapter Number	Research Motivations
	physical manipulations required to form the woven reinforcement on to the tool. In this chapter, a review on how existing drape simulators have been used to produce layup instructions is presented. The verdict of this review was that current methods have failed to generate this sequence of physical manipulations. In this chapter, a novel solution to this bottleneck is introduced in the form of deep reinforcement learning with the primary aim of modelling the human learning process during hand lamination, which was lacking in existing drape simulators.
6	A preliminary case study was carried out using a ramp mould shape to test the reinforcement learning based solution introduced in chapter 5. The modelling procedure used for this case study is summarized in this chapter.
7	Results from the reinforcement learning based preliminary case study are summarized in this chapter. While results from this case study suggested that the proposed novel solution is effective in producing manufacturing instructions for hand laminators, it was identified that further case studies that are based on more complex mould shapes are required to further support this claim.
8	In this chapter, two more case studies are performed using more complex 3D mould shapes to further investigate the working mechanics of the proposed reinforcement learning based solution. Results and findings from these two case studies are summarized in this chapter.
9	In this chapter, conclusions and potential future work in relation to all the research presented in this thesis are discussed.

(Appendix: Table 1 continued)

Appendix: Table 2 - Scoring criteria used for the layup of a composite flat panel

Action Number	Scoring Criteria	Score
1	A lab coat was worn, and this action was performed immediately after entering the clean room.	1
2	Latex gloves were worn on both hands, and this action was performed immediately after Action 1.	1
3	All relevant tools (2 L shapes, Stanley Knife and roller) were picked up and placed on the table, and this action was performed after Action 2.	1
4	The stack of prepreg was picked up from the defrosting station and placed on the table, and this action was performed after Action 3.	1
5	The first layer, marked 0^o, was aligned accurately using the 2 L shapes and this action was performed after Action 4.	1
6	The Stanley knife was used to remove the top backing of the first layer in the fibre direction, and this action was performed after Action 5.	1
7	A tick was drawn in front of the first layer marked as 0^o on the piece of paper provided, and this action was performed after Action 6.	1
8	The Stanley knife was used to remove the bottom backing of the second layer in the fibre direction, and this action was performed after Action 7.	1
9	The provided piece of release film was placed on top of the first layer with a small gap, and this action was performed after Action 8.	1

Action Number	Scoring Criteria	Score
10	The second layer was laid on top of the first layer with alignment checks and gradual release film pull outs, and this action was performed after Action 9.	1
11	First two layers were consolidated using the roller without removing the top backing of the second layer, and this action was performed after Action 10.	1
12	The Stanley knife was used to remove the top backing of the second layer in the fibre direction, and this action was performed after Action 11.	1
13	A tick was drawn in front of the second layer marked as 90^{o} on the piece of paper provided, and this action was performed after Action 12.	1
14	Actions 8-13 are repeated for layer 3 in the correct order.	6
15	Actions 8-13 are repeated for layer 4 in the correct order.	6
16	Laid up four layers were moved to the consolidation station, and this action was performed after Action 15.	1
17	The provided piece of release film was placed on top of the four layers, and this action was performed after Action 16.	1
18	A preformed frame was picked up and placed on top of the four layers, and this action was performed after Action 17.	1

(Appendix: Table 2 continued)

Action Number	Scoring Criteria	Score
19	A vacuum pump was connected to the frame, and this action was performed after Action 18.	1

(Appendix: Table 2 continued)

Appendix: Table 3 - All possible omission patterns for the action sequence: Action 50, Action 36, Action 34, Action 34, Action 34. (Each row represents an action sequence and omitted actions are indicated in red bold letters).

Action 50	Action 36	Action 34	Action 34	Action 34
Action 50	**Action 36**	Action 34	Action 34	Action 34
Action 50	Action 36	**Action 34**	Action 34	Action 34
Action 50	Action 36	Action 34	**Action 34**	Action 34
Action 50	Action 36	Action 34	Action 34	**Action 34**
Action 50	**Action 36**	Action 34	Action 34	Action 34
Action 50	Action 36	**Action 34**	Action 34	Action 34
Action 50	Action 36	Action 34	**Action 34**	Action 34
Action 50	Action 36	Action 34	Action 34	**Action 34**
Action 50	**Action 36**	**Action 34**	Action 34	Action 34

Action 50	**Action 36**	Action 34	**Action 34**	Action 34
Action 50	**Action 36**	Action 34	Action 34	**Action 34**
Action 50	Action 36	**Action 34**	**Action 34**	Action 34
Action 50	Action 36	**Action 34**	Action 34	**Action 34**
Action 50	Action 36	Action 34	**Action 34**	**Action 34**
Action 50	**Action 36**	**Action 34**	Action 34	Action 34
Action 50	**Action 36**	Action 34	**Action 34**	Action 34
Action 50	**Action 36**	Action 34	Action 34	**Action 34**
Action 50	Action 36	**Action 34**	**Action 34**	Action 34
Action 50	Action 36	**Action 34**	Action 34	**Action 34**
Action 50	Action 36	Action 34	**Action 34**	**Action 34**
Action 50	**Action 36**	**Action 34**	**Action 34**	Action 34
Action 50	**Action 36**	**Action 34**	Action 34	**Action 34**
Action 50	**Action 36**	Action 34	**Action 34**	**Action 34**
Action 50	Action 36	**Action 34**	**Action 34**	**Action 34**
Action 50	**Action 36**	**Action 34**	**Action 34**	Action 34

(Appendix: Table 3 continued)

Action 50	Action 36	Action 34	Action 34	Action 34
Action 50	Action 36	Action 34	Action 34	Action 34
Action 50	Action 36	Action 34	Action 34	Action 34
Action 50	Action 36	Action 34	Action 34	Action 34

(Appendix: Table 3 continued)

Appendix: Table 4 - Scoring criteria for the clean room layup (Refer to Figure 4.3 for accurate grip styles and hand gestures, refer to Table 4.1 for action descriptions)

Action	Scoring Criteria	Total Score
1	- Accurate grip of prepreg - Datum met	2
2	- Accurate right hand gesture - Accurate left hand gesture - Accurate direction of pressure application - Action performed	4
3	- Accurate right hand gesture - Accurate left hand gesture - Accurate direction of pressure application - Action performed - Correct order of actions 2 and 3	5
4	- Accurate left hand gesture - Accurate Dibber grip - Accurate direction of pressure application	4

Action	Scoring Criteria	Total Score
	- Action performed	
5	- Accurate left hand gesture - Accurate Dibber grip - Accurate direction of pressure application - Action performed - Correct order of actions 4 and 5	5
6	- Accurate right hand gesture - Accurate left hand gesture - Accurate direction of pressure application - Action performed	4
7	- Accurate right hand gesture - Accurate left hand gesture - Accurate direction of pressure application - Action performed - Correct order of actions 6 and 7	5
8	- Accurate right hand gesture - Accurate left hand gesture - Action performed	3
9	- Accurate left hand gesture - Accurate Dibber grip - Accurate direction of pressure application - Action performed	4
10	- Accurate left hand gesture - Accurate Dibber grip - Accurate direction of pressure application	5

(Appendix: Table 4 continued)

Action	Scoring Criteria	Total Score
	- Action performed - Correct order of actions 9 and 10	
11 - 16	Not scored separately since these actions are identical to actions 8-10, but performed on a different location on the mould.	0
17	- Accurate right hand gesture - Accurate left hand gesture - Accurate direction of pressure application - Action performed	4
18	- Accurate right hand gesture - Accurate left hand gesture - Accurate direction of pressure application - Action performed - Correct order of actions 17 & 18	5
19	- Accurate Dibber grip - Accurate direction of pressure application - Action Performed	3

(Appendix: Table 4 continued)

Additional appendix material is provided in digital format saved in a CD with this thesis. Contents of this CD are summarised below:

1. Unity 3D project file of the VR simulator for the layup of a unidirectional prepreg flat panel (Refer to Chapter 3 for more information);
2. Unity 3D project file of the VR simulator for the layup of a 3D mould with a sheet of woven prepreg as described in Chapter 4.
3. Unity 3D project file and relevant MATLAB scripts for the AI layup agent as explained in Chapters 5,6 and 7.